

**SISTEM KENDALI SUDUT *PITCH PROJECTOR HEADLAMP*  
SEPEDA MOTOR BERBASIS *CLOSED LOOP SYSTEM***

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun oleh:

Cipto Bagus Jati Kusumo

NIM. 145150300111048



PROGRAM STUDI TEKNIK KOMPUTER  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

### PENGESAHAN

SISTEM KENDALI SUDUT PITCH PROJECTOR HEADLAMP SEPEDA MOTOR BERBASIS CLOSED LOOP SYSTEM

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun Oleh :

Cipto Bagus Jati Kusumo  
NIM. 145150300111048

Skripsi ini telah diuji dan dinyatakan lulus pada  
4 Mei 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dahnil Syauqy, S.T., M.T., M.Sc.  
NIP: 2016078704231002

Dosen Pembimbing II

Wijaya Kurniawan, S.T., M.T.  
NIP: 198201252015041002

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D  
NIP: 19710518 200312 1 001

## IDENTITAS TIM PENGUJI

### 1. Penguji 1, Ketua Majelis

Nama : Rizal Maulana, S.T., M.T., M.Sc.

NIK : 2016078910091001

### 2. Penguji 2

Nama : Mochammad Hannats Hanafi Ichsan, S.ST, M.T

NIK : 201405 881229 1 001



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, didalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik disuatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 28 Maret 2018

Cipto Bagus Jati Kusumo  
NIM: 145150300111048





## PLAGIASI (TURNITIN)

### Skripsi

#### ORIGINALITY REPORT

<b>5%</b>	<b>4%</b>	<b>0%</b>	<b>3%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

#### PRIMARY SOURCES

<b>1</b>	<b>Submitted to Universitas Brawijaya</b> Student Paper	<b>1%</b>
<b>2</b>	<b>www.hukumonline.com</b> Internet Source	<b>1%</b>
<b>3</b>	<b>www.ifac-papersonline.net</b> Internet Source	<b>&lt;1%</b>
<b>4</b>	<b>www.scribd.com</b> Internet Source	<b>&lt;1%</b>
<b>5</b>	<b>Submitted to University of Westminster</b> Student Paper	<b>&lt;1%</b>
<b>6</b>	<b>Submitted to Swinburne University of Technology</b> Student Paper	<b>&lt;1%</b>
<b>7</b>	<b>Submitted to Asian Institute of Technology</b> Student Paper	<b>&lt;1%</b>
<b>8</b>	<b>repository.usu.ac.id</b> Internet Source	<b>&lt;1%</b>
<b>9</b>	<b>mechatronics2004.newcastle.edu.au</b>	

## DAFTAR RIWAYAT HIDUP

### DATA PRIBADI

1. Nama Lengkap : Cipto Bagus Jati Kusumo
2. Tempat dan Tanggal Lahir : Situbondo, 7 Desember 1995
3. Kewarganegaraan : Indonesia
4. Agama : Islam
5. Alamat : Jalan Dinoyo Permai No. 71 Kec. Lowokwaru Malang
6. Status : Belum Kawin
7. Surat Elektronik : [cbagusjk@live.com](mailto:cbagusjk@live.com)

### PENDIDIKAN FORMAL

No.	Jenjang Pendidikan	Nama Sekolah	Kab/Kota	Tahun
1	Taman Kanak – Kanak	TK Nurul Mansyur	Situbondo	2001 - 2002
2	Sekolah Dasar	SDN 1 Curah Jeru	Situbondo	2002 – 2008
3	Sekolah Menengah Pertama	SMPN 1 Situbondo	Situbondo	2008 – 2011
4	Sekolah Menengah Akhir	SMAN 1 Situbondo	Situbondo	2011 – 2014
5	Strata 1 (Satu)	Universitas Brawijaya	Malang	2014 – 2018

### PENDIDIKAN NON FORMAL

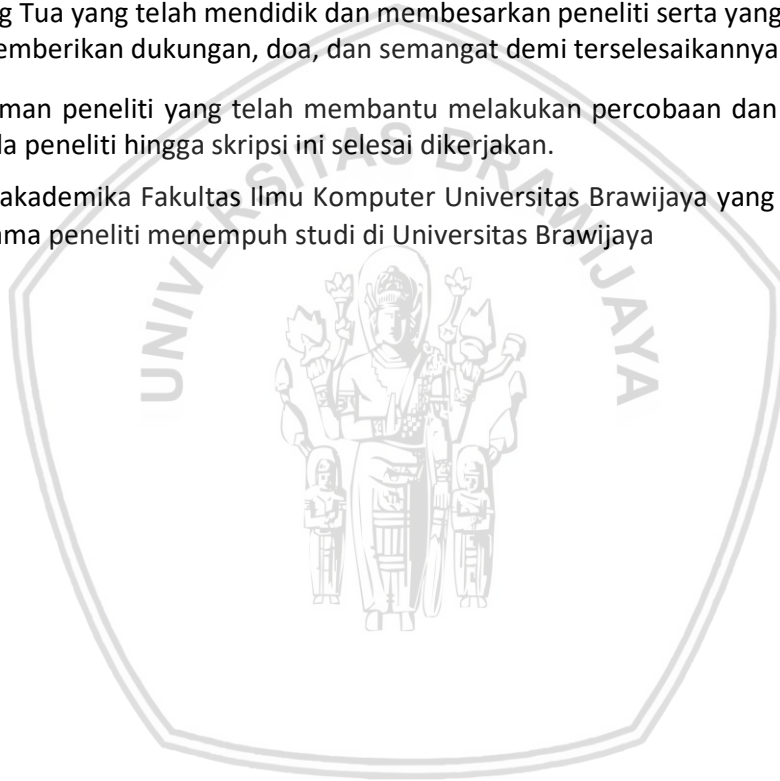
No.	Nama	Lembaga Pelaksana	Tahun
1	Diklat Jurnalistik	SMAN 1 Situbondo	2011
2	PK2MABA Universitas Brawijaya	Universitas Brawijaya	2014
3	PK2MABA Fakultas Ilmu Komputer	BEM Fakultas Ilmu Komputer	2014
4	Pelatihan Anggota Robotiik	Robotiik Fakultas Ilmu Komputer	2011 – 2014

## UCAPAN TERIMA KASIH

Peneliti menyadari semua pengerjaan dan penyusunan laporan skripsi ini tidaklah akan berhasil tanpa bantuan dari berbagai pihak, oleh karena itu peneliti ingin mengucapkan dan menyampaikan rasa hormat beserta terima kasih kepada :

1. Bapak Dahnia Syauqy, S.T., M.T., M.Sc. dan Bapak Wijaya Kurniawan, S.T., M.T. selaku dosen pembimbing skripsi yang telah meluangkan waktu pikiran dan tenaga untuk membantu membimbing dan mengarahkan peneliti sehingga skripsi ini dapat terselesaikan.
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika.
3. Kedua Orang Tua yang telah mendidik dan membesarkan peneliti serta yang tiada henti – hentinya memberikan dukungan, doa, dan semangat demi terselesaikannya skripsi ini.
4. Teman – teman peneliti yang telah membantu melakukan percobaan dan memberikan saran kepada peneliti hingga skripsi ini selesai dikerjakan.

Seluruh civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak membantu selama peneliti menempuh studi di Universitas Brawijaya



## ABSTRAK

Lampu utama penerangan kendara roda dua pada negara Indonesia ini umumnya menggunakan gas Halogen yang hasil dari penerangannya kurang untuk jalanan yang minim PJU (Penerangan Jalan Umum), dengan kreatifitas pengguna kendaraan tersebut maka sumber penerangan diganti dengan sumber yang menghasilkan penerangan lebih. Seperti contoh lampu yang menggunakan gas Xenon atau biasa dikenal dengan *High Intensity Dishcharge* Xenon, namun dengan penggantian ini muncul masalah baru yaitu pancaran cahayanya liar yang menyilaukan pengguna jalan lain. Penggunaan *Headlamp Projector* adalah salah satu cara untuk menghindari pancaran cahaya liar tersebut yang menyilaukan tersebut dengan cara memfokuskan hanya pada jalan raya saja, pancaran cahaya yang fokus tersebut haruslah diatur agar fokusnya tidak mengarah kepada pengguna jalan lainnya. Oleh karena itu untuk mempermudah pengemudi dalam mengatur fokus pancaran cahaya, dibangunlah sistem ini.

Sistem bekerja dengan bantuan sensor *IMU MPU6050* beserta ke 6 sudut bacaannya yang terdapat pada *headlamp* sepeda motor untuk dapat membaca sudut *pitch* saat ini, aktuator unit yang berupa stepper motor beserta mekanisme aktuasinya dan metode pengontrol *Closed Loop System*. Nilai sudut saat ini yang dibaca oleh sensor kemudian diselihkan dengan nilai set poin yang telah dikalkulasi sebelumnya, hasil dari selisih tersebut kemudian dijadikan acuan oleh sistem untuk mengontrol aktuator yang akan mempengaruhi sudut *pitch headlamp*. Tingkat keberhasilan dari sistem ini adalah 98,154 % dari 6 kali percobaan pengatur ketinggian pancaran cahaya dengan beban penumpang pada sepeda motor yang berbeda – beda.

Kata kunci : *Levelling Headlamp, Headlamp Projector, High Intensity Discharge, Pitch Projector, Closed Loop System*.

## ABSTRACT

*The main lights of the motorcycle illumination in the country of Indonesia generally use Halogen gas which results from less illumination for the minimal road PJU (Street Lighting), with the creativity of the vehicle users then the source of lighting replaced with sources that produce more light. Like the example of lights that use Xenon gas or commonly known as High Intensity Dishcharge Xenon, but with the replacement of this, new problem emerging, light emitting wildly dazzling other road users. The use of the Headlamp Projector is the only one to avoid the dazzling glare of the wild light by focusing only on the road, the beam of focusing light must be set to road so that the focus does not lead to other road users. Therefore, this system is built to facilitate the driver in setting the focus of the light beam.*

*The system works with the help of MPU6050 IMU sensor with all 6 axis sensing on the motorcycle headlamp in order to read the current pitch angle value, actuator unit including the machanism, and controll metode that is Closed Loop System. The current angle value read by the sensor is then set aside with the value of the set points that have been in the previous calculation, the result of the difference is then made reference by the system to control the actuator that will affect the pitch angle of the headlamp. The success rate of this system is 98.154% of the six attempts to adjust the altitude of light beam with different passenger loads on the same motorcycle.*

**Keyword :** *Levelling Headlamp, Headlamp Projector, High Intensity Discharge, Pitch Projector, Closed Loop System.*

## KATA PENGANTAR

Puji syukur kehadiran Tuhan YME yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga pengerjaan dan penyusunan laporan skripsi yang berjudul *"SISTEM KENDALI SUDUT PITCH PROJECTOR HEADLAMP SEPEDA MOTOR BERBASIS CLOSED LOOP SYSTEM"* ini dapat terselesaikan.

Peneliti menyadari semua pengerjaan dan penyusunan laporan skripsi ini tidaklah akan berhasil tanpa bantuan dari berbagai pihak, oleh karena itu peneliti ingin mengucapkan dan menyampaikan rasa hormat beserta terima kasih kepada :

5. Bapak Dahnial Syauqy, S.T., M.T., M.Sc. dan Bapak Wijaya Kurniawan, S.T., M.T. selaku dosen pembimbing skripsi yang telah meluangkan waktu pikiran dan tenaga untuk membantu membimbing dan mengarahkan peneliti sehingga skripsi ini dapat terselesaikan.
6. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika.
7. Kedua Orang Tua yang telah mendidik dan membesarkan peneliti serta yang tiada henti – hentinya memberikan dukungan, doa, dan semangat demi terselesaikannya skripsi ini.
8. Teman – teman peneliti yang telah membantu melakukan percobaan dan memberikan saran kepada peneliti hingga skripsi ini selesai dikerjakan.
9. Seluruh civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak membantu selama peneliti menempuh studi di Universitas Brawijaya.

Seluruh penyusunan skripsi ini pastinya belum lah sempurna dan masih banyak yang perlu dibenahi, oleh karena itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata peneliti berharap bahwa hasil penelitian ini dapat berguna dan bermanfaat bagi semua pihak yang menggunakannya.

Malang, 28 Maret 2018

Peneliti  
cbagusjk@live.com

## DAFTAR ISI

PENGESAHAN .....	i
IDENTITAS TIM PENGUJI .....	ii
PERNYATAAN ORISINALITAS .....	iii
PLAGIASI (TURNITIN) .....	iv
DAFTAR RIWAYAT HIDUP .....	v
UCAPAN TERIMA KASIH .....	vi
ABSTRAK .....	vii
ABSTRACT .....	viii
KATA PENGANTAR .....	ix
DAFTAR ISI .....	x
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR .....	xiv
DAFTAR LAMPIRAN .....	xvi
BAB 1 PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan .....	3
1.4. Manfaat .....	3
1.5. Batasan Masalah .....	3
1.6. Sistematika Pembahasan .....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1. Tinjauan Pustaka .....	6
2.2. Dasar Teori .....	6
2.2.1. Metode <i>Close Loop Control System</i> .....	7
2.2.2. Sudut <i>Pitch</i> .....	8
2.2.3. <i>Headlamp Projector</i> .....	8
2.2.4. Sensor Unit Pengukur Inersia : GY521 MPU6050 .....	9
2.2.5. <i>Stepper Motor</i> : 28BYJ-48 5V .....	10
2.2.6. <i>Driver Motor DC</i> : L293D .....	10
2.2.7. Kontroler Unit : Arduino Nano ATmega328P .....	11
BAB 3 METODOLOGI .....	12
3.1. Langkah – Langkah Penelitian .....	12



3.1.1.	Pre – Riset.....	13
3.1.2.	Studi Literatur .....	13
3.1.3.	Tahap Perancangan.....	14
3.1.4.	Tahap Implementasi Purwarupa .....	14
3.1.5.	Tahap Pengujian dan Analisis .....	14
3.1.6.	Kesimpulan .....	14
3.2	Metode .....	14
3.2.1.	Perancangan Mekanisme dan Elektronika Sistem .....	15
3.2.2.	Pengumpulan Komponen Elektronika Sistem .....	15
3.2.3.	Desain Mekanisme Sistem .....	16
3.2.4.	Implementasi Mekanisme dan Elektronika Sistem .....	16
3.2.5.	Perancangan Perangkat Lunak Sistem .....	16
3.2.6.	Implemenasi Perangkat Lunak Sistem .....	16
BAB 4	REKAYASA KEBUTUHAN .....	17
4.1.	Gambaran Umum Sistem .....	17
4.1.1.	Perspektif Sistem .....	17
4.1.2.	Ruang Lingkup.....	18
4.1.3.	Asumsi Ketergantungan .....	18
4.2.	Kebutuhan Sistem .....	18
4.2.1.	Kebutuhan Fungsional .....	18
4.2.2.	Kebutuhan Non Fungsional .....	19
4.2.3.	Kebutuhan Perangkat Keras .....	19
4.2.4.	Kebutuhan Perangkat Lunak .....	21
BAB 5	PERANCANGAN DAN IMPLEMENTASI .....	22
5.1.	Perancangan .....	22
5.1.1.	Perancangan Perangkat Keras: <i>Power Supply Unit</i> .....	22
5.1.2.	Perancangan Perangkat Keras: <i>Driver</i> , Kontroler Unit dan Aktuator .....	23
5.1.3.	Perancangan Perangkat Keras: Sensor dan Kontroler Unit .....	25
5.1.4.	Perancangan Perangkat Keras: Mekanisme Aktuasi .....	26
5.1.5.	Perancangan Perangkat Keras: Peletakan Komponen Sistem .....	26
5.1.6.	Perancangan Perangkat Lunak .....	30
5.2.	Implementasi .....	47
5.2.1.	Implementasi Perangkat Keras: <i>Power Supply Unit</i> .....	48
5.2.2.	Implementasi Perangkat Keras: Sensor .....	48

5.2.3.	Implementasi Perangkat Keras: Kontroler Unit .....	49
5.2.4.	Implementasi Perangkat Keras: Aktuator .....	49
5.2.5.	Implementasi Perangkat Lunak .....	50
BAB 6 PENGUJIAN .....		55
6.1.	Parameter Pengujian .....	55
6.2.	Skenario Pengujian .....	55
6.3.	Pengujian Pembacaan Sensor .....	56
6.3.1.	Tujuan Pengujian .....	56
6.3.2.	Prosedur Pengujian .....	56
6.3.3.	Hasil Pengujian .....	57
6.3.4.	Analisa Pengujian .....	57
6.4.	Pengujian Kesalahan Sistem .....	58
6.4.1.	Tujuan Pengujian .....	58
6.4.2.	Prosedur Pengujian .....	58
6.4.3.	Hasil Pengujian .....	58
6.4.4.	Analisa Hasil Pengujian .....	59
6.5.	Pengujian Waktu .....	60
6.5.1.	Tujuan Pengujian .....	60
6.5.2.	Prosedur Pengujian .....	60
6.5.3.	Hasil Pengujian .....	61
6.5.4.	Analisa Pengujian .....	61
BAB 7 PENUTUP .....		63
7.1.	Kesimpulan .....	63
7.2.	Saran .....	64
DAFTAR PUSTAKA .....		65

## DAFTAR TABEL

Tabel 5.1 Koneksi Antar Pin <i>Driver</i> , Kontroler Unit dan Aktuator .....	24
Tabel 5.2 Koneksi Antar Pin Sensor dan Kontroler Unit.....	25
Tabel 5.3 <i>Noise Accelerometer</i> .....	43
Tabel 5.4 Konfigurasi Sensor .....	50
Tabel 5.5 Konfigurasi <i>Timer 1</i> .....	50
Tabel 5.6 Konfigurasi Set Poin .....	51
Tabel 5.7 Meminta Data Sensor .....	51
Tabel 5.8 Menyimpan Data Sensor .....	51
Tabel 5.9 Mengelola Data Sensor .....	52
Tabel 5.10 <i>Summing Junction</i> .....	52
Tabel 5.11 Batas Toleransi Error .....	52
Tabel 5.12 <i>Interrupt Service Routine</i> .....	53
Tabel 5.12 (lanjutan) .....	54
Tabel 6.1 Hasil Pengujian Pembacaan Sensor .....	57
Tabel 6.2 Hasil Pengujian Kesalahan Sistem pada Permukaan Datar .....	59
Tabel 6.3 Hasil Pengujian Waktu pada Permukaan Datar .....	61

## DAFTAR GAMBAR

Gambar 2.1 <i>Closed Loop dan Open Loop</i> .....	7
Gambar 2.2 <i>Yaw Pitch Roll</i> .....	8
Gambar 2.3 <i>Headlamp Projector</i> .....	9
Gambar 2.4 <i>Breakboard GY521</i> .....	9
Gambar 2.5 <i>Stepper Motor 28BYJ-48 5V</i> .....	10
Gambar 2.6 <i>Driver L293D</i> .....	11
Gambar 2.7 <i>Arduino Nano ATmega328P</i> .....	11
Gambar 3.1 <i>Langkah – Langkah Penelitian</i> .....	12
Gambar 3.2 <i>Urutan Pembuatan Sistem</i> .....	15
Gambar 4.1 <i>Blok Diagram Sistem Secara Keseluruhan</i> .....	17
Gambar 5.1 <i>Power Supply Unit</i> .....	23
Gambar 5.2 <i>Driver, Kontroler Unit dan Aktuator</i> .....	24
Gambar 5.3 <i>Koneksi Sensor dan Kontroler Unit</i> .....	25
Gambar 5.4 <i>Mekanisme Aktuasi</i> .....	26
Gambar 5.5 <i>Peletakan Perangkat Keras Sistem</i> .....	26
Gambar 5.6 <i>Contoh Jarak Antar Sudut</i> .....	27
Gambar 5.7 <i>Skenario Percobaan</i> .....	28
Gambar 5.8 <i>Nilai Sudut Bacaan Sensor</i> .....	29
Gambar 5.9 <i>Tinggi Pancaran Cahaya pada Tembok</i> .....	29
Gambar 5.10 <i>Mencari Sudut Headlamp / <math>\angle ADF</math></i> .....	29
Gambar 5.11 <i>Alur Instruksi Sistem</i> .....	30
Gambar 5.12 <i>Register PWR_MGMT_1</i> .....	31
Gambar 5.13 <i>Register GYRO_CONFIG</i> .....	33
Gambar 5.14 <i>Register ACCEL_CONFIG</i> .....	34
Gambar 5.15 <i>Register TCCR1A</i> .....	35
Gambar 5.16 <i>Register TCCR1B</i> .....	35
Gambar 5.17 <i>Register TIMSK1</i> .....	37
Gambar 5.18 <i>Ilustrasi Pencarian Sudut Pitch (<math>\alpha</math>)</i> .....	38
Gambar 5.19 <i>Tataletak dan Jarak Komponen Pengujian Sistem</i> .....	39
Gambar 5.20 <i>Predefined Process Meminta dan Memproses Data Sensor</i> .....	41
Gambar 5.21 <i>Sudut Pitch Alpha 0,833</i> .....	44
Gambar 5.22 <i>Sudut Pitch Alpha 0,98</i> .....	44
Gambar 5.23 <i>Predefined Process Plant Signal (Interrupt Service Routine)</i> .....	46

Gambar 5.23 (lanjutan) .....	47
Gambar 5.24 Perangkat Keras : <i>Power Supply Unit</i> .....	48
Gambar 5.25 Perangkat Keras : Sensor .....	48
Gambar 5.26 Perangkat Keras : Kontroler Unit .....	49
Gambar 5.27 Perangkat Keras : Aktuator .....	49
Gambar 6.1 Pengujian .....	55



## DAFTAR LAMPIRAN

LAMPIRAN A KODE PROGRAM SISTEM .....	67
--------------------------------------	----



## BAB 1 PENDAHULUAN

Pada bab ini akan menjabarkan tentang latar belakang masalah yang diangkat menjadi topik penelitian serta metode yang akan digunakan, pertanyaan – pertanyaan penelitian untuk mengarahkan penelitian, tujuan penelitian, manfaat penelitian, batasan masalah penelitian serta sistematika pembahasan.

### 1.1. Latar Belakang

Penggunaan lampu utama pada kendaraan bermotor yang menyilaukan sangat menyakitkan mata dan membahayakan pengguna jalan lain, hal ini terjadi karena lampu utama yang digunakan telah diganti ke lampu utama yang dayanya lebih besar, pengaturan ketinggian lampu yang salah atau penggunaan reflektor yang tidak sesuai dengan lampu itu sendiri sehingga mengakibatkan banyaknya pancaran sinar liar (sinar yang tidak fokus/*glare*) yang membahayakan pengguna jalan lain (AAAFoundation, 2012).

Kasus kecelakaan lalu lintas terbanyak di Indonesia terjadi pada jenis kendaraan sepeda motor dengan jumlah 31.270 kasus lalu diikuti oleh mobil dengan jumlah 6.268 kasus pada tahun 2017, jika dilihat dari jenis kecelakaannya maka paling banyak terjadi akibat berpapasan dengan pengguna jalan lainnya sebanyak 6.530 kasus (Polantas, 2017). Walaupun jenis kecelakaan tersebut tidak semuanya dikarenakan pancaran lampu yang menyilaukan namun sebagai tindakan dini maka penelitian ini bisa dijadikan titik awal pencegahan terhadap maraknya penggunaan lampu yang menyilaukan.

Dari data kasus kecelakaan lalu lintas yang mencakup masalah dalam penelitian ini yaitu pancaran sinar lampu utama pada kendaraan bermotor terutama sepeda motor yang telah diganti ke daya yang lebih besar (Contoh lampu jenis *High Intensity Discharge/HID*), maka solusi untuk mengatasi hal tersebut adalah menggunakan *Headlamp Projector* atau *Poliellipsoidal* yang memproyeksikan sinar dari reflektor khusus ke jalan namun dengan *cutoff* atau batasan cahaya untuk meminimalisir sinar liar/*glare* yang dapat menyilaukan pengguna jalan lainnya (Dastern.torque.net, 2014).

Namun penggunaan *Headlamp Projector* atau *Poliellipsoidal* saja tidak cukup karena jika beban pada bagian belakang pengemudi bertambah seperti contoh ada barang bawaan atau pembonceng, maka sudut  $x$  (*Pitch*) akan berubah menjadi lebih besar yang berakibat naiknya pancaran sinar *Headlamp Projector* atau *Poliellipsoidal*.

Masalah pertama pengaturan sudut  $x$  (*Pitch*) meskipun bisa diatur secara manual pada reflektor namun akan menyulitkan pengemudi karena pengemudi harus membawa engkol untuk melonggarkan baut pengaturnya, hal ini tentu merepotkan bahkan setelah mengatur ketinggian pancaran lampu kendaraan karena ukuran engkol pun tidak akan muat ditempatkan pada kantong pakaian pengguna dan akan memakan waktu yang cukup lama. Masalah kedua adalah



pembonceng atau beban pada bagian belakang sepeda motor harus ditempatkan terlebih dahulu kemudian pengaturan ketinggian pancaran lampu baru bisa dilakukan karena akan memberikan hasil yang tepat sesuai dengan Peraturan Pemerintah Nomor 55 Tahun 2012 Pasal 24 Ayat 2 yang berbunyi “Untuk Sepeda Motor harus dilengkapi dengan lampu utama dekat dan lampu utama jauh paling banyak dua buah dan dapat memancarkan cahaya paling sedikit 40 (empat puluh) meter ke arah depan untuk lampu utama dekat dan 100 (seratus) meter ke arah depan untuk lampu utama jauh.” (Indonesia, 2012) jika ketinggian lampu diatur dengan adanya beban yang terpasang. Berdasarkan masalah – masalah pengaturan ketinggian lampu dan efek jika salah mengatur ketinggian pancaran lampu, maka dari itu dibutuhkan sistem yang dapat mengontrol sudut  $x$  (*Pitch*) secara otomatis tanpa campur tangan pengemudi.

Sistem yang dapat mengontrol suatu objek secara otomatis merupakan sistem kendali yang memerlukan suatu metode untuk mengendalikan objek tersebut yang dalam penelitian ini adalah *Headlamp Projector* atau *Poliellipsoidal*. Sistem ini berisi komponen – komponen seperti mikrokontroller unit untuk memproses informasi masukan dan menjadikannya sinyal pada aktuator, sensor inersia sebagai indra sistem yang memberikan informasi masukan sesuai dengan ruang lingkup pengendalian yang berupa 3 sudut dari *Accelerometer* dan 3 sudut dari *Gyroscope*, serta aktuator sebagai pemberi aktuasi kepada objek yang akan mempengaruhi masukan informasi selanjutnya yang berupa *Stepper Motor*. Penggunaan metode *Close Loop System* ini dipilih karena dapat melakukan pengontrolan lebih cepat dibandingkan dengan metode sejenis seperti PID, hal ini dapat dilihat dari struktur metodenya pada masing – masing kontroler. *Close Loop System* hanya perlu melakukan operasi pengurangan pada data umpan balik dengan data set poin (Electronicshub, 2015), sedangkan *PID* masih membutuhkan operasi perkalian dengan *Gain* untuk tiap kontroler *P*, *I*, dan *D*. Karena ada 3 kontroler pada *PID* maka ada 3 kalkulasi berbeda yang berjalan bergantian yaitu kontroler *P*, kontroler *I* yang harus melakukan operasi Integral, dan *D* yang melakukan operasi Turunan (Instruments, 2011).

Kebutuhan akan pengontrol sudut *Headlamp Projector* atau *Poliellipsoidal* dibuktikan dengan adanya regulasi internasional UNECE ( *United Nation Economic Commission for Europe* ) *Regulation 48* (UNECE, 2014).

Sistem untuk mengontrol sudut  $x$  (*Pitch*) ini sangat dibutuhkan karena sebagai pengendara kendaraan bermotor kita haruslah fokus dalam berkendara serta memastikan bahwa keadaan kendaraan yang digunakan dalam kondisi baik. Hal ini terkait pula dengan pancaran sinar lampu utama yang harus tidak menyilaukan pengguna jalan lain namun tetap menyajikan visibilitas yang maksimal.

## 1.2. Rumusan Masalah

Pada bagian ini akan memuat pertanyaan penelitian untuk mengarahkan penelitian, mendorong peneliti untuk menjawabnya dan menarik minat pembaca.

1. Bagaimana sistem dapat mengetahui nilai sudut  $x$  (*Pitch*) saat ini?

2. Bagaimana sistem dapat mengontrol nilai sudut  $x$  (*Pitch*) saat ini agar sesuai dengan nilai yang benar ?
3. Bagaimana implementasi metode *Closed – Loop System* ?
4. Bagaimana performansi sistem berdasarkan pengontrol *Closed – Loop System* ?

### 1.3. Tujuan

Sub bab ini berisi tujuan yang ingin dicapai dari penelitian ini, berikut adalah tujuan – tujuan penelitian ini.

1. Sistem dapat mengetahui nilai sudut *Pitch* saat ini.
2. Sistem dapat mengontrol nilai *Pitch* saat ini agar sesuai dengan nilai sudut yang baik.
3. Mengetahui implementasi metode *Close – Loop System*.
4. Mengetahui performansi sistem berdasarkan pengontrol *Close – Loop System*.

### 1.4. Manfaat

Penelitian ini juga mempunyai manfaat yang ingin dicapai, berikut adalah manfaat yang diinginkan dari penelitian ini.

1. Diharapkan dapat mencegah atau meminimalisir terjadinya kecelakaan lalu lintas akibat pancaran sinar lampu utama yang menyilaukan.
2. Mempermudah pengemudi dalam mengatur ketinggian dari *Headlamp Projector* atau *Poliellipsoidal*.

### 1.5. Batasan Masalah

Bagian ini berisi tentang poin – poin yang digunakan untuk membantu menjelaskan ruang lingkup masalah penelitian dengan menyatakan hal – hal yang menjadi batasan dan asumsi – asumsi yang digunakan untuk menyelesaikan masalah yang sudah dirumuskan.

1. Implementasi purwarupa pada sepeda motor merk Yamaha tipe *Sport* nama *New Vixion Lighting* yang menggunakan *Projector* pada *Headlampnya*.
2. Sudut yang dikontrol ada 1 yaitu sudut  $x$  (*Pitch*).
3. Masukan data pada sistem untuk menentukan nilai *Pitch* yang benar adalah masukan dari peneliti yang nantinya akan dijadikan acuan oleh sistem.
4. Sistem bekerja hanya saat sepeda motor tidak bergerak dan pada kondisi permukaan datar. Selama sistem mengatur ketinggian dari pancaran lampu, sepeda motor tidak diperkenankan untuk bergerak karena akan mempengaruhi hasil.

5. Jika Sistem telah selesai mengatur ketinggian pancaran lampu, maka sistem harus dipadamkan lalu sepeda motor dapat bergerak.

## 1.6. Sistematika Pembahasan

### BAB 1 PENDAHULUAN

Pada BAB 1 dijelaskan permasalahan yang menjadi topik penelitian, rumusan masalah, tujuan dari penelitian, manfaat penelitian, batasan masalah dari penelitian dan sistematika pembahasan dari Sistem Kendali Sudut *Pitch Projector Headlamp* Berbasis *Closed Loop System*.

### BAB 2 LANDASAN KEPUSTAKAAN

BAB 2 berisi tentang ringkasan penelitian – penelitian yang berkaitan atau satu bidang dengan penelitian yang akan dilakukan oleh peneliti, kemudian dijelaskan juga dasar – dasar teori dari metode yang akan digunakan, sudut *pitch*, *projector headlamp*, serta perangkat keras sistem seperti mikrokontroler dan sensor yang digunakan.

### BAB 3 METODOLOGI

Metodologi berkaitan dengan tahapan – tahapan dan tata cara melakukan penelitian, hal ini dimulai dari preriset, studi literatur, tahapan perancangan, tahapan implementasi purwarupa sampai pengambilan kesimpulan.

### BAB 4 REKAYASA KEBUTUHAN

Pada BAB ini menjelaskan tentang kebutuhan – kebutuhan sistem baik secara perangkat keras seperti *power supply* unit, kontroler unit / mikrokontroler, sensor, *driver* untuk membantu kontroler unit mengontrol aktuator, serta aktuator itu sendiri. Kemudian kebutuhan perangkat lunak seperti *bootloader* pada mikrokontroler, *library* yang digunakan, serta program *Integrated Development Environment (IDE)*. Selain itu ada kebutuhan fungsional dan non fungsional.

### BAB 5 PERANCANGAN DAN IMPLEMENTASI

BAB ini menjelaskan tentang perancangan sistem dari segi perangkat keras yang berupa desain skematik berdasarkan kebutuhan perangkat keras seperti skematik dari *power supply unit* menuju ke keseluruhan sistem, skematik dari kontroler unit ke *driver* dan kontroler unit ke aktuator, serta skematik dari kontroler unit ke sensor. Kemudian terdapat juga desain dari mekanisme aktuator dan peletakan komponen sistem pada headlamp. Untuk dari segi perangkat lunak tersedia *flowchart* yang menjelaskan alur dari program kemudian dilanjutkan dengan implementasi dari desain sistem dan mekanik baik dari segi perangkat keras maupun perangkat lunak.

## BAB 6 PENGUJIAN

Pengujian merupakan langkah terakhir dari penelitian sebelum akhirnya mengambil kesimpulan dan memberikan kritik dan saran, pada BAB ini akan berisi parameter pengujian, skenario pengujian, serta pengujian kesalahan atau *error* dan pengujian waktu aktuasi yang juga dijelaskan tujuan pengujian, prosedur pengujian hasil pengujian dan analisa pengujian ada tiap-tiap pengujian.

## BAB 7 PENUTUP

Berisi Kesimpulan yang menjawab pertanyaan dari masalah penelitian serta diangkat dari hasil pengujian yang dilakukan, selain kesimpulan juga berisi saran yang diberikan dari peneliti untuk pengembangan selanjutnya.

## DAFTAR PUSTAKA

Seluruh dokumen – dokumen yang dirujuk dalam proposal ini tercatat pada daftar pustaka.



## BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan memaparkan salah satu dari beberapa bagian penting dalam penelitian yaitu tinjauan pustaka yang berisi penelitian – penelitian dengan menggunakan metode yang sama, dasar teori yang berisi teori dasar dari elemen – elemen yang akan digunakan pada sistem.

### 2.1. Tinjauan Pustaka

Pada bagian ini akan memaparkan beberapa penelitian – penelitian yang menggunakan metode yang sama dengan penelitian ini. Paparan dari penelitian – penelitian tersebut dapat memberikan gambaran bahwa metode yang digunakan dalam penelitian ini sudah pernah digunakan, diuji, dan didapatkan hasilnya.

Peter Opgen-Rhein<sup>1)</sup>, Torsten Bertam<sup>2)</sup>, Jorgen Seuss<sup>3)</sup>, Peter Karas<sup>3)</sup>, dan Dieter Strychik<sup>3)</sup> dari University of Duisburg-Essen<sup>1)</sup>, Ilmenau Technical University<sup>2)</sup>, Hella KG Huek & Co.<sup>3)</sup> dalam publikasinya yang berjudul *A HARDWARE-IN-THE-LOOP BASED PROCESS IMPROVES QUALITY AND DECREASES THE DEVELOPMENT PERIOD OF A DYNAMIC HEADLAMP LEVELLING SYSTEM* menjelaskan bahwa sistem pengatur ketinggian pancaran cahaya (*Headlamp Levelling System*) dapat dicapai dengan mendapatkan sudut *Pitch* dari badan kendaraan dan aktuatornya dapat menggunakan *Stepper Motor*. (Peter Opgen-Rhein, 2004).

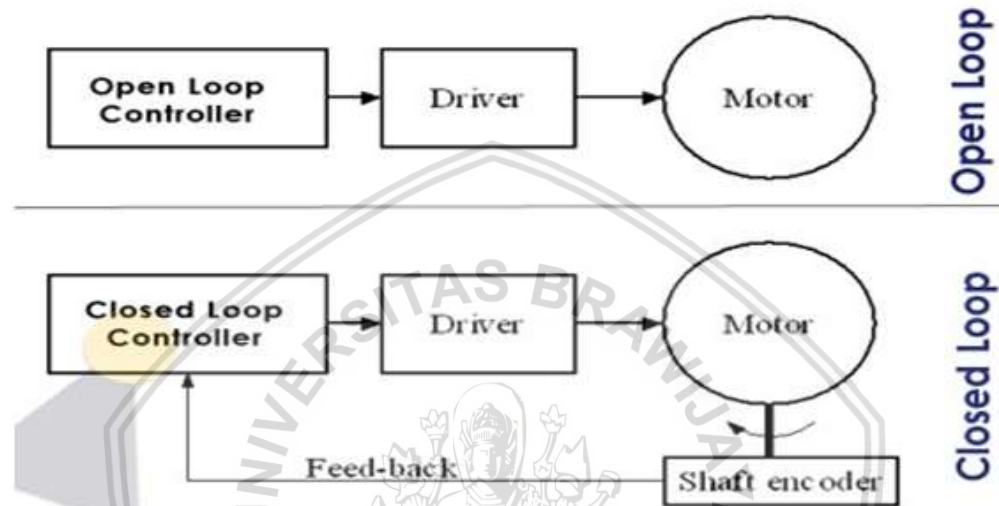
S.Karaman, T. Hacibekir, E. Kural, E.S. Öztürk, M. Demirci and B. Aksun Güvenç dari Istanbul Technical University dalam *Papernya* yang berjudul *DESIGN AND HARDWARE-IN-THE-LOOP SIMULATION OF AN AUTOMATIC HEADLIGHT CONTROL SYSTEM* mempresentasikan *hardware in the loop simulation of An Adaptive Headlight System for Motor Vehicles. Headlight Orientation Control System* mengatur sisi kanan dan kiri *Low Beam* tiap *Headlight* secara individual, memiliki otomatisasi *Headlamp Levelling System* sebagai kompensasi terhadap adanya perubahan beban pada kendaraan bermotor, serta dapat menghilangkan efek *Pitching* saat kendaraan pertama kali berakselerasi atau saat pengereman untuk visibilitas yang lebih baik pada malam hari. Dari hasil penelitian ini didapatkan kesimpulan bahwa sistem yang didesain mampu memperhitungkan sudut referensi yang optimal untuk *Headlight* sesuai dengan jalanan pada beberapa kondisi (S. Karaman, 2006).

### 2.2. Dasar Teori

Pada bagian ini akan menjelaskan teori dasar dari elemen – elemen yang akan digunakan oleh sistem dalam penelitian ini. Untuk menggunakan suatu elemen maka peneliti harus terlebih dahulu mengetahui dasar teorinya guna mempermudah dalam mengetahui karakteristik dan tingkah laku dari elemen yang akan digunakan.

### 2.2.1. Metode *Close Loop Control System*

Kendali terhadap suatu sistem elektronik terutama mekatronik yang berisi perangkat keras mekanikal serta elektronik pada umumnya terdiri dari 2 macam tipe yaitu *Open Loop* dan *Close Loop*. Perbedaan diantara keduanya adalah ada tidaknya umpan balik atau *Feedback* dari hasil aktuasi atau keluaran saat ini terhadap masukan informasi berikutnya, untuk lebih jelasnya akan digambarkan pada Gambar 2.1.



Gambar 2.1 *Closed Loop dan Open Loop*

(Nasir, 2014)

Pada Gambar 2.1 terlihat jelas bahwa perbedaan mendasar adalah *feedback*, dan untuk mentranslasikan informasi dari hasil aktuasi ke informasi masukan yang harus berupa sinyal digital maka diperlukan sensor. Sensor berperan aktif dalam kontroler tipe ini karena secara terus menerus harus memberikan informasi tambahan untuk dijadikan masukan, bilamana sensor tidak bekerja dengan benar maka metode kontroler ini tidak akan merespon terhadap perubahan yang terjadi pada objek yang dikontrolnya.

Bagian dalam dalam kotak *Closed Loop Controller* pada Gambar 2.1 telah terdapat nilai referensi yang telah ditetapkan oleh peneliti atau biasa disebut dengan *Set Point* dan *Summing Point* yang mengolah sinyal masukan tambahan dari sensor dan *Set Point*. *Summing Point* adalah proses selisih antara *Set Point* dengan informasi masukan tambahan, hasilnya adalah *Error* yang akan dikontrol oleh sistem melalui aktuator agar nilai *Error* menjadi 0 (Electronicshub, 2015).

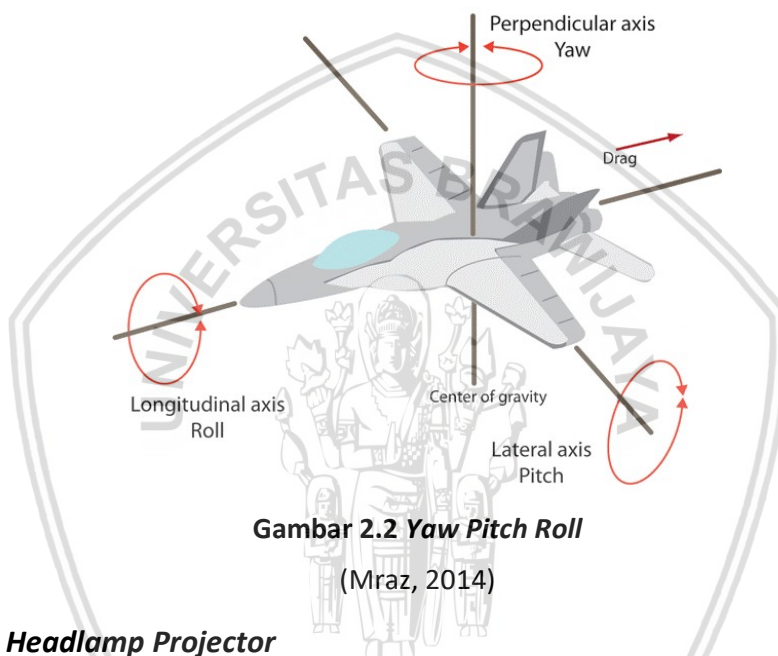
Metode ini tidak menggunakan banyak *resource*, cepat diproses serta cocok untuk tipe mikrokontroler yang umum tersedia dipasaran.



### 2.2.2. Sudut *Pitch*

*Pitch* adalah pergerakan dari sudut  $x$ . Selayaknya kendaraan yang dapat bergerak bebas 3 sudut, seperti pesawat tempur atau kapal selam yang dapat mengubah ketinggian dan rotasinya. Ketiga sudut orthogonal itu adalah longitudinal, horizontal, dan vertikal. Untuk lebih memudahkan pemahaman, lihat pada Gambar 2.2 (Mraz, 2014).

Sudut *Pitch* adalah sudut yang mempengaruhi naik turunnya hidung pada pesawat, sudut *Roll* adalah sudut yang mempengaruhi kemiringan kekiri atau kekanan dari pesawat sedangkan sudut *Yaw* adalah sudut yang mempengaruhi arah dari pesawat.



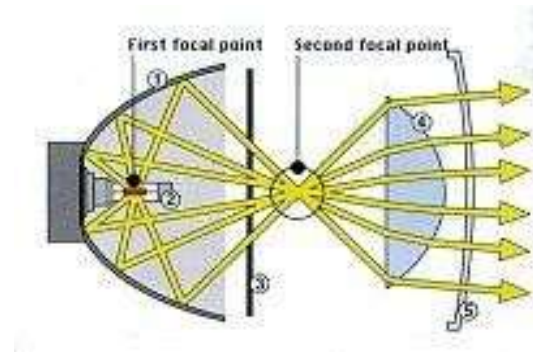
**Gambar 2.2 Yaw Pitch Roll**  
(Mraz, 2014)

### 2.2.3. *Headlamp Projector*

*Headlamp Projector / Ellipsoidal Reflector* adalah salah satu tipe lampu utama yang menggunakan *Projector* untuk memproyeksikan sinarnya. *Ellipsoidal Reflector* merefleksikan sinar dari lampu utama ke *second focal point*, kemudian ada *Shields* yang bertindak sebagai pembatas pendistribusian sinar dari *second focal point* agar sinar tidak sepenuhnya keluar (*cut off*) lalu sinar yang telah dibatasi tersebut diproyeksikan ke jalan dengan lensa datar cembung (Dastern.torque.net, 2014)

*Headlamp Projector* terdapat banyak tipe dipasaran namun yang digunakan untuk penelitian ini adalah tipe *MiniH1* keluaran pabrikan AES yang memang didesain khusus untuk digunakan oleh lampu HID (*High Intensity Discharge*) dengan tipe *fitting* H1. Daya yang dianjurkan untuk *headlamp projector* ini adalah 35 Watt. Untuk detail dari bagian – bagian *headlamp projector* dapat dilihat pada Gambar 2.3.





**Gambar 2.3 Headlamp Projector**

(Dastern.torque.net, 2014)

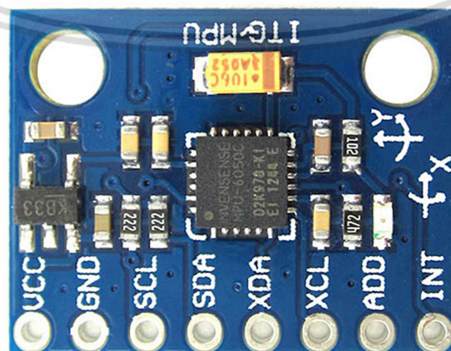
Keterangan :

1. Reflector
2. Light Source
3. Shield
4. Lens
5. Cover

#### 2.2.4. Sensor Unit Pengukur Inersia : GY521 MPU6050

Unit Pengukur Inersia atau biasa disebut *IMU* Sensor adalah sebuah sensor yang berisi *Gyroscope* Sensor dan/atau *Accelerometer* (jika 6-axis). Didalam perangkat semikonduktor sensor ini terdapat *microelectromechanical system* yang dapat mengukur *angular velocity*.

Pada penelitian ini Sensor unit pengukur inersia yang digunakan adalah MPU6050 yang terdapat pada *breakboard* GY521. Sensor ini telah terdapat *Gyroscope* serta *Accelerometer* yang menjadi satu dalam sirkuit terpadu sehingga menjadikannya lebih praktis, Gambar 2.4 berikut adalah *breakout board* GY521.



**Gambar 2.4 Breakboard GY521**

(Geekbuying, 2018)

Sensor ini mempunyai kemampuan untuk membaca *angular velocity* dengan *Gyroscope*, terdapat beberapa cakupan/*range* yang dapat diatur dengan mengakses *register – register* pada MPU6050 dari cakupan yang sempit sehingga hanya dapat membaca *angular velocity* yang pelan (maksimal 250 derajat/detik) hingga cakupan yang luas (maksimal 2000 derajat/detik) (Invensense, 2017).

Selain *Gyroscope* adapula *Accelerometer* yang juga mempunyai cakupan – cakupan dari  $\pm 2g$  sampai  $\pm 16g$ . Data yang dikirim ke kontroler unit pun menggunakan komunikasi *Serial* tipe *I2C* yang hanya menggunakan 2 buah jalur dengan *clock speed* 400 KHz. Untuk tegangan yang digunakan adalah 3.3 Volt, tegangan tersebut didapat berkat regulator yang telah dibenamkan pada *breakout board* GY521 yang meregulasi tegangan 5 Volt dari kontroler unit menjadi 3.3 Volt.

#### 2.2.5. Stepper Motor : 28BYJ-48 5V

*Stepper* motor adalah salah satu tipe motor arus searah yang telah ditentukan perputarannya sebanyak sekian derajat bergantung pada jumlah fasa serta rasio giginya.



**Gambar 2.5 Stepper Motor 28BYJ-48 5V**

(Electronics, 2018)

Seperti namanya yaitu *Stepper* Motor, motor ini dapat mengubah sinyal listrik menjadi gerakan dengan *Step* per *Step* yang dikontrol oleh *Driver*. *Step* untuk motor ini adalah sinyal kotak atau *Square Wave* dengan frekuensi 100 Hz, jika kurang dari itu maka motor akan berputar terlalu lama dan jika lebih dari 100 Hz ada kemungkinan koil tidak akan terinduksi dengan baik yang mengakibatkan motor tidak berputar. Selain itu urutan dari *perstep*nya harus diperhatikan, namun dari pabrik sudah diurutkan sesuai dengan urutan kabel pada pin yang tersedia (Stateham, 2016)

#### 2.2.6. Driver Motor DC : L293D

Untuk mengontrol *Stepper* Motor diharuskan menggunakan *Driver* atau kontroler tambahan karena kontroler unit atau mikrokontroler tidak dilengkapi dengan pengaman untuk komponen induksi. *Driver* L293D merupakan *driver* yang umum dan mudah digunakan.



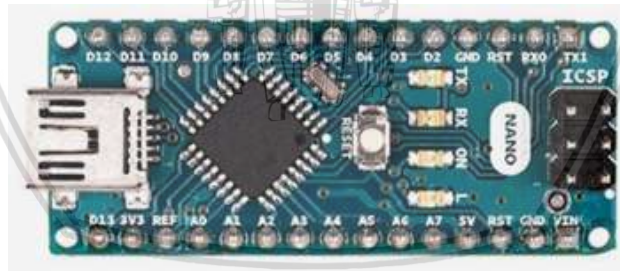
**Gambar 2.6 Driver L293D**

(Instruments T. , 2018)

*Package* yang dipilih untuk digunakan adalah DIP16, *package* ini berisi 16 pin yang dapat dengan mudah ditancapkan pada *development board*. *Driver* ini dapat mengontrol Motor DC dengan tegangan hingga 36 Volt dan arus maksimal 600 mA, sedangkan untuk masukan dari kontroler unit hanya perlu tegangan 5 Volt. Untuk menjalankan *driver* ini cukup dengan memberikan sinyal kotak dari kontroler unit dengan urutan sesuai dengan stepper motor yang akan digunakan (Instruments T. , 2018)

#### 2.2.7. Kontroler Unit : Arduino Nano ATmega328P

Kontroler Unit yang digunakan adalah Mikrokontroler ATmega328P yang benamkan pada *development board* Arduino Nano. ATmega328P adalah SoC (*System on Chip*) yang didalamnya telah terdapat CPU (*Central Processing Unit*) berbasis AVR dengan arsitektur RISC, 32 KiloBytes Flash Memory *ReWrite* dan *ReRead Complatible*, 1 KiloByte EEPROM, 2 KiloBytes RAM, 23 GPIO.



**Gambar 2.7 Arduino Nano ATmega328P**

(Store.Arduino, 2018)

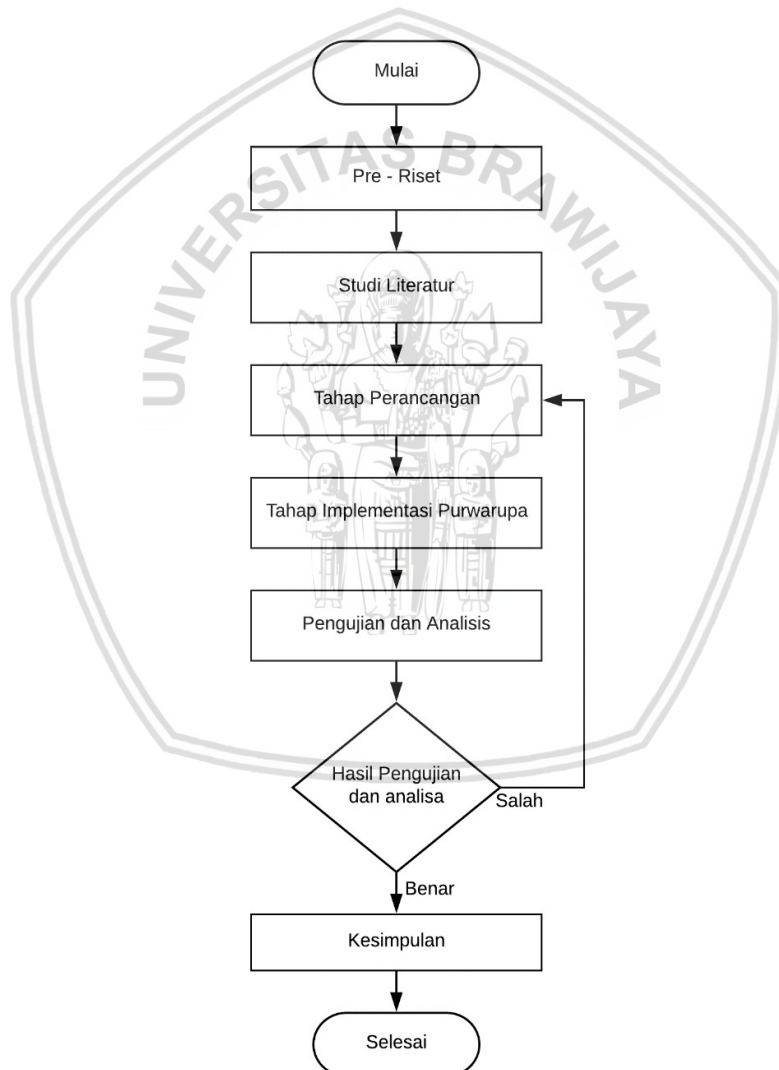
Untuk menggunakan kontroler unit ini dibutuhkan aplikasi untuk membuat program yaitu *Arduino IDE* yang sudah terdapat pada halaman Arduino dan dapat diunduh dengan gratis. Selain sisi perangkat lunak, pada sisi perangkat keras Arduino Nano ini dapat diberi masukan tegangan hingga 12 Volt, hal ini akan diregulasi oleh regulator Arduino Nano hingga menjadi 5 Volt yang menjadi tingkat tegangan yang aman digunakan oleh kontroler unit ATmega328P.

## BAB 3 METODOLOGI

Pada BAB ini akan membahas tentang langkah – langkah penelitian serta langkah – langkah dalam pembuatan sistem yang akan dikerjakan pada penelitian ini.

### 3.1. Langkah – Langkah Penelitian

Pada bagian ini akan menjelaskan langkah – langkah penelitian, langkah – langkah tersebut sangat penting karena terdapat alur yang jelas tentang apa saja yang akan dilakukan selama penelitian, Gambar 3.1 menjelaskan langkah – langkah penelitian.



Gambar 3.1 Langkah – Langkah Penelitian

### 3.1.1. Pre – Riset

Tahap pertama ini dipersiapkan tentang hal – hal yang dibutuhkan untuk melakukan penelitian seperti perangkat keras dan metode untuk memecahkan masalah penelitian. Adapun hal – hal tersebut sebagai berikut :

1. Metode Penyelesaian Masalah Penelitian.

Pada bagian ini dilakukan pencarian metode yang pas dengan masalah yang diangkat pada penelitian ini. Penulis menggunakan metode *Close Loop System* karena penggunaan metode ini tidak memerlukan *resources* yang banyak, selain itu dapat dengan mudah digunakan pada model sistem mikrokontroler yang banyak beredar dipasaran dan relatif mudah didapatkan seperti contoh mikrokontroler ATmega328P.

2. Pemilihan Sensor Yang Sesuai dengan Keperluan Data Masukan.

Sesuai dengan permasalahan penelitian yang diangkat yaitu berkaitan dengan nilai sudut  $x$  (*Pitch*), maka sensor yang berkaitan erat dengan bidang ini adalah MPU6050 (*Gyroscope* dan *Accelerometer*). Sensor ini dapat mengukur *angular velocity* yang nantinya dapat dijadikan data masukan untuk diproses di mikrokontroler lalu hasil pemrosesannya dikirimkan ke aktuator.

3. Pemilihan Kontroler Unit yang Sesuai.

Kontroler Unit yang sesuai dengan masalah penelitian yaitu Kontroler Unit yang secara ukuran kecil karena akan diimplementasikan pada sepeda motor, harga jual tidak mahal, umum digunakan agar lebih mudah didapat dipasaran, dapat menerima informasi dari sensor dan memprosesnya sesuai dengan metode yang digunakan serta hasil prosesnya dapat dijadikan sinyal untuk aktuator. Dalam penelitian ini penulis menggunakan mikrokontroler yaitu ATmega328P.

4. Pemilihan Aktuator.

Aktuator yang berhubungan dengan bidang pada permasalahan penelitian ini adalah aktuator yang dapat berkatuasi memutar untuk menggerakkan pengatur ketinggian cahaya, dalam hal ini adalah suatu motor. Motor pun harus dapat bergerak memutar 2 arah dengan kecepatan yang sama dan harus dapat berhenti dengan cepat bila diperintahkan oleh kontroler unit, selain itu motor harus dapat berputar dengan rotasi tak terbatas. Peneliti memilih stepper motor dc karena masuk dalam kriteria yang diperlukan.

### 3.1.2. Studi Literatur

Studi Literatur dilakukan untuk dapat menggunakan metode / perangkat lunak hingga perangkat keras yang akan digunakan untuk penelitian ini. Studi literatur seperti penggunaan metode *Close Loop System* dilakukan berdasarkan panduan – panduan buku atau artikel jurnal yang memuat tata cara penggunaan metode ini dengan benar. Begitu pula dengan penggunaan perangkat keras lainnya.



### **3.1.3. Tahap Perancangan**

Perancangan sistem adalah tahap dimana sistem dirancang secara perangkat lunak ( program ) dan perangkat keras ( desain alat dan elektrikal ), perancangan secara perangkat keras dilakukan terlebih dahulu karena informasi data yang akan diproses bergantung kepada desain dari perangkat keras lalu kemudian dirancang algoritma pemecahan masalah penelitian secara perangkat lunak.

### **3.1.4. Tahap Implementasi Purwarupa**

Implementasi Purwarupa dilakukan setelah perancangan perangkat lunak dan perangkat keras dilakukan. Implementasi perangkat lunak dilakukan dengan membuat program yang akan menentukan perilaku sistem sedangkan Implementasi perangkat keras dilakukan dengan membuat purwarupa dari desain sistem sesuai perancangan dan merangkai komponen elektronik purwarupa.

### **3.1.5. Tahap Pengujian dan Analisis**

Tahap implementasi purwarupa telah selesai maka dilakukan pengujian dan analisis yang bertujuan untuk menguji apakah masalah penelitian sudah benar – benar terselesaikan atau belum, jika belum kembali lagi ke tahap perancangan dengan tujuan untuk merivisi rancangan sistem baik secara perangkat keras ataupun perangkat lunak sesuai dengan masalah yang muncul.

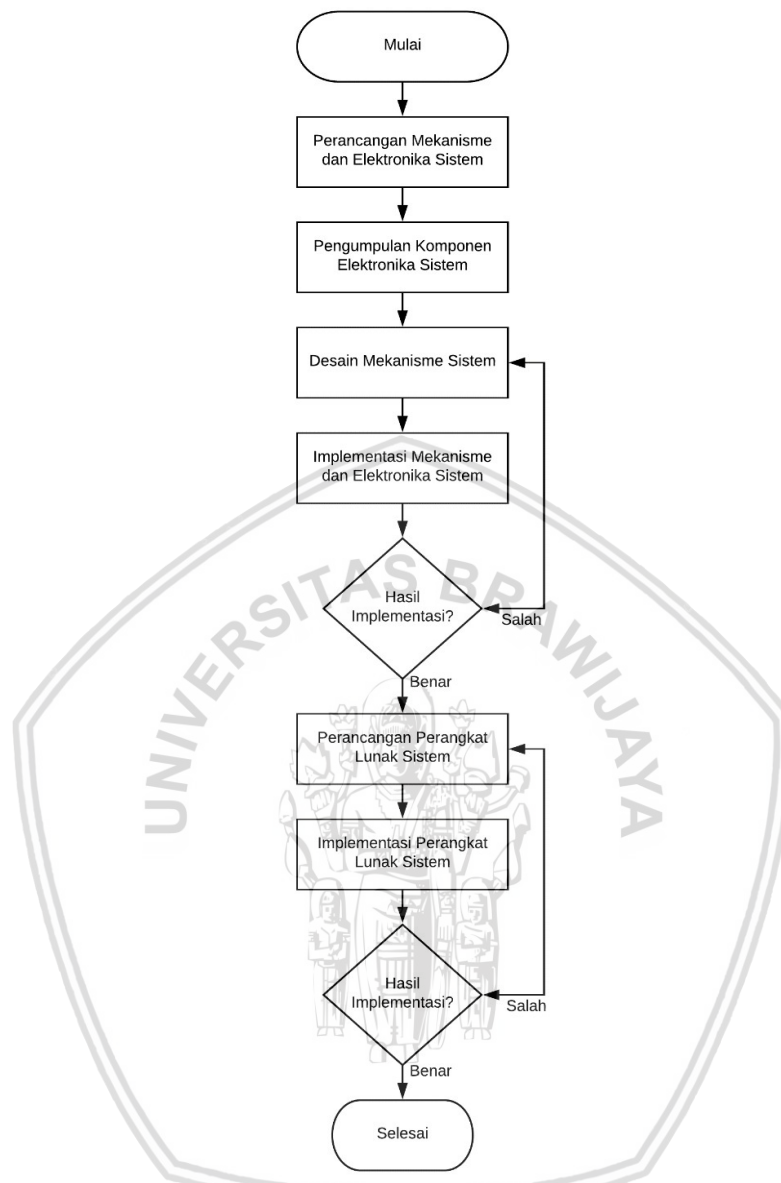
### **3.1.6. Kesimpulan**

Pengambilan kesimpulan dilakukan berdasarkan hasil analisis dari keadaan sistem.

## **3.2 Metode**

Pada bagian ini akan membahas tentang urutan pembuatan sistem baik dari segi perangkat keras maupun perangkat lunak. Pembuatan ini terdiri dari beberapa tahap seperti Perancangan Mekanisme dan Elektronika, Pengumpulan Komponen Elektronika, Desain Mekanisme Sistem, Implementasi Mekanisme dan Elektronika Sistem, Perancangan Perangkat Lunak, dan Implementasi Perangkat Lunak.

Perancangan Perangkat Keras terlebih dahulu dikerjakan karena penempatan – penempatan dari komponen elektronika dan mekanisme dari sistem akan mempengaruhi Perangkat Lunak. Setelah Perangkat Keras selesai diimplementasikan kemudian Perangkat Lunak dirancang dan diimplementasikan. Gambar 3.2 pada halaman 16 akan menggambarkan urutan pembuatan sistem.



**Gambar 3.2 Urutan Pembuatan Sistem**

### 3.2.1. Perancangan Mekanisme dan Elektronika Sistem

Perancangan Mekanisme dan Elektronika Sistem dilakukan pada awal – awal pembuatan sistem karena garis besar dari sistem harus dibuat untuk menjadikan acuan Desain Mekanisme dan Implementasi Mekanisme dan Elektronika Sistem. Pada tahap ini diperoleh rancangan secara garis besar dari sistem yang akan dibuat.

### 3.2.2. Pengumpulan Komponen Elektronika Sistem

Setelah rancangan mekanisme dan elektronika selesai, maka dilakukan pengumpulan komponen elektronika sistem dengan cara mencari referensi secara *online* / pada toko – toko *online* serta referensi secara *offline* / pada toko – toko



fisik yang tersedia di daerah Kota Malang dan Kota Surabaya. Pengumpulan komponen elektronika ini penting karena ada bagian dari Mekanisme Sistem yang bergantung pada ukuran dan bentuk komponen elektronika.

### **3.2.3. Desain Mekanisme Sistem**

Proses Desain Mekanisme Sistem dilakukan dengan cara pengukuran langsung pada komponen – komponen sistem seperti *headlamp projector*, pada motor dc geared, serta *servo* motor untuk membuat desain yang pas. Selain pengukuran, dilakukan juga proses mendapatkan bentuk fisik dengan cara menempelkan kertas ke bagian yang ingin didapatkan bentuknya lalu diberikan sedikit tekanan pada area yang diinginkan lalu hasilnya dipindai oleh mesin pemindai sehingga menghasilkan gambar yang dapat diolah pada program komputer. Hal ini dilakukan karena pengukuran sukar dilakukan pada area – area tertentu.

### **3.2.4. Implementasi Mekanisme dan Elektronika Sistem**

Implementasi Mekanisme dan Elektronika Sistem dilakukan setelah hasil desain dari Mekanisme Sistem dicetak sehingga membentuk fisik seperti pada Desain Mekanisme Sistem yang dibuat.

### **3.2.5. Perancangan Perangkat Lunak Sistem**

Setelah Perangkat Keras selesai diimplementasikan, maka selanjutnya dilakukan perancangan Perangkat Lunak yang berdasarkan Perangkat Keras yang sudah dibuat. Perancangan ini berupa *flowchart* yang menggambarkan garis besar dari alur program.

### **3.2.6. Implementasi Perangkat Lunak Sistem**

Implementasi Perangkat Lunak Sistem adalah tahap akhir dari urutan pembuatan sistem, Perangkat Lunak dibuat pada program khusus lalu diunduh kekontroler unit untuk kemudian dijalankan oleh kontroler unit.

## BAB 4 REKAYASA KEBUTUHAN

Bab 4 Rekayasa Kebutuhan akan membahas mengenai prasyarat – prasyarat atau kebutuhan – kebutuhan dari sistem yang dibangun agar bisa memenuhi tujuan yang telah dibuat sebelumnya. Kebutuhan – kebutuhan tersebut meliputi Gambaran Umum Sistem, Kebutuhan Fungsional, Kebutuhan Perangkat Keras, dan Kebutuhan Perangkat Lunak.

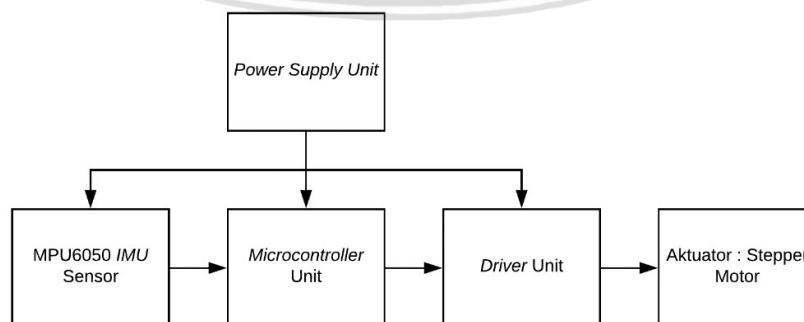
### 4.1. Gambaran Umum Sistem

Sistem ini bekerja mengontrol sudut *Pitch* dari *Headlamp Projector* kendaraan roda dua, masukan data berupa data mentah *accelerometer* dan *gyroscope* dari *IMU Sensor MPU6050* yang terdiri dari *3-axis Accelerometer* dan *3-axis Gyroscope* dengan total *6-axis*. *MPU6050* ditempatkan pada *Headlamp Projector* dengan keadaan horizontal sudut *Pitch*nya dipengaruhi oleh badan kendaraan roda dua.

Data masukan tersebut diproses oleh sebuah *Microcontroller Unit* berbasis *ATMega328P* dengan *Bootloader Open Source Arduino Nano*, pemrosesan yang dimaksud meliputi meminta data mentah dari sensor, memproses data mentah menjadi data dengan satuan derajat, mengelola data tersebut dengan metode *Closed Loop System* yang mengontrol aktuator berupa *Stepper Motor* untuk sudut *Pitch*.

#### 4.1.1. Perspektif Sistem

Sistem ini ditujukan untuk pengguna kendaraan roda dua yang menggunakan *Headlamp Projector*, sistem ini akan membantu pengemudi dalam mengatur ketinggian pancaran cahaya dari *Headlamp Projector* agar tidak mengganggu pengendara lain. Untuk blok diagram secara keseluruhan dari sistem terdapat pada Gambar 4.1 berikut, pada sisi sebelah kiri terdapat sensor *MPU6050* dan sebelah atas terdapat *power supply unit* yang merupakan masukan, *microcontroller unit* dan *driver unit* yang merupakan unit pemroses, serta aktuator yang merupakan keluaran.



Gambar 4.1 Blok Diagram Sistem Secara Keseluruhan

#### 4.1.2. Ruang Lingkup

Sistem ini dibangun untuk digunakan oleh pengendara roda dua yang ingin melakukan *Retrofit* agar mempermudah dalam mengatur arah pancaran sinar dari lampu utama, ketentuan yang digunakan dalam sistem ini adalah ketinggian kendaraan dari tanah setinggi 78,8 centimeter. Selama ini pengaturan arah pancaran sinar dari lampu utama dilakukan dengan *manual* yang berarti pengendara harus mengatur arah pancaran sinar dari lampu utama tiap kali ada perubahan sudut pada badan kendaraan agar arah pancaran sinar dari lampu utama aman bagi pengendara lain, maka dari itu sistem ini dibuat agar dapat membantu pengguna kendaraan roda dua untuk mengatur arah pancaran sinar dari lampu utama dengan otomatis.

#### 4.1.3. Asumsi Ketergantungan

Asumsi ketergantungan yang terdapat pada sistem ini adalah

1. Penempatan Lampu Utama maksimal 78,8 centimeter dari tanah.
2. Reflektor lampu utama merupakan *Headlamp Projector* yang memiliki pembatas cahaya.

#### 4.2. Kebutuhan Sistem

Kebutuhan sistem merupakan semua kebutuhan yang diperlukan agar sistem dapat bekerja sesuai dengan tujuan, berikut adalah kebutuhan sistem :

##### 4.2.1. Kebutuhan Fungsional

Kebutuhan fungsional diperlukan untuk sistem dapat menjalankan tugasnya, berikut adalah kebutuhan fungsional :

1. Kontroler Unit yang dapat mengatur sensor, meminta, menerima dan menyimpan data masukan mentah.

Kontroler Unit mengirimkan perintah mengakses *register* dari sensor via komunikasi *Serial I2C* untuk mengatur sensor yang meliputi: *Power Managment*, *Full Scale Range Accelerometer*, *Full Scale Range Gyroscope*, serta *Digital Low Pass Filter*. Kemudian kontroler unit mengirimkan *request* untuk meminta data mentah dari sensor yang akan disimpan pada tempat yang telah disediakan pada kontroler unit.

2. Kontroler Unit yang dapat memproses data masukan mentah menjadi data masukan yang memiliki satuan derajat.

Kontroler unit memproses data mentah dari *Accelerometer* dengan teori Trigonometri dan Pitagoras serta *Gyroscope* sesuai dengan panduan pada *datasheet* MPU6050.

3. Kontroler Unit yang dapat mengelola data masukan dengan metode *Close Loop System*.

Data masukan dikelola dengan metode *Close Loop System*. Hal ini merupakan hal utama dari sistem karena sinyal keluaran dari metode ini akan digunakan untuk mengontrol aktuator yang nantinya akan berpengaruh pada sudut *Pitch Headlamp Projector*.

4. Kontroler Unit yang dapat mengontrol aktuator berdasarkan hasil dari pengolahan data masukan dengan metode *Close Loop System*.

Hasil dari pengolahan data digunakan untuk mengontrol aktuator berupa Stepper Motor. Aktuator memberikan aktuasi pada *Headlamp Projector* yang telah disematkan sensor MPU6050 agar dapat berada pada sudut yang diinginkan.

#### 4.2.2. Kebutuhan Non Fungsional

1. Sensor Unit Pengukur Inersia

Unit pengukur inersia mampu membaca perubahan percepatan sudut minimal 250 derajat/detik, mampu membaca arah gravitasi, mampu bekerja pada tingkat tegangan 5 Volt, dan dapat berkomunikasi via *Serial I2C / USART / SPI*.

2. Kontroler Unit

Kontroler Unit dapat bekerja pada tegangan 12 Volt, dapat berkomunikasi dengan perangkat lain via *Serial I2C / USART / SPI*, dapat diprogram dengan bahasa C++ / Java, dan terdapat GPIO untuk mengontrol perangkat – perangkat lain yang tidak berkomunikasi via *Serial*.

3. Aktuator

Aktuator dapat bergerak 2 arah serta dapat diatur kecepatan putarnya dan mampu berhenti dengan cepat jika diperintahkan oleh kontroler unit.

4. *Driver Unit*

Dapat mentranslasikan sinyal dari kontroler unit ke perangkat lain dan melindungi kontroler unit dari masalah elektronik induksi.

5. *Power Supply Unit*

*Power Supply Unit* dapat meregulasi tegangan dari 12 Volt menjadi 5 Volt dengan arus minimal 500 mA.

#### 4.2.3. Kebutuhan Perangkat Keras

Sistem Tertanam membutuhkan perangkat keras untuk dapat bekerja mencapai tujuan yang diinginkan, berikut adalah kebutuh perangkat keras :

1. *Power Supply Unit*

*Power Supply Unit* atau PSU merupakan unit untuk memberikan daya pada semua perangkat keras pada sistem, PSU meregulasi tegangan masukan menjadi tegangan yang dapat diterima oleh sistem yang pada hal ini merupakan modul *buck-converter* berbasis IC LM2596 yang menurunkan tegangan dari 12 Volt

menjadi 5 Volt dengan maksimal arus 3 A untuk digunakan oleh kontroler unit, sensor IMU, *driver* unit, serta stepper motor.

## 2. Kontroler Unit

Kontroler unit merupakan *System On Chip* yang digunakan untuk memproses data masukan mentah hingga mengontrol aktuator, berbasis ATmega328P yang umum digunakan dan juga merupakan varian *Pico Power* yang lebih hemat daya.

## 3. Sensor IMU

GY521 merupakan *breakout board* yang berbasis IC sensor IMU InvenSense MPU6050 dengan 6-axis dalam satu IC yang membuatnya lebih mudah ditempatkan karena fisiknya yang kecil, sensor ini mampu memberikan masukan yang dibutuhkan oleh sistem.

## 4. Driver Unit

*Driver* unit digunakan untuk membantu kontroler unit dalam mengontrol aktuator *stepper* motor. *Driver* unit yang digunakan berbasis IC L293D yang dapat menerima tegangan kontrol hingga 36 Volt, selain itu IC L293D memiliki *internal flyback diode* untuk proteksi sistem *internal* dan kontroler unit dari *spike* induksi.

## 5. Aktuator

Aktuator merupakan pemberi aktuasi pada sistem yang dikontrol, Stepper Motor 28BYJ-48 5V adalah pilihan yang tepat karena selain mudah digunakan, mudah didapatkan, pergerakannya cukup halus dengan *stride angle*  $0,09^\circ$  tiap *step* dalam mode *driving half step*. Motor ini juga dapat bergerak 2 arah dan dapat berhenti dengan cepat.

Mengacu pada *datasheet*, motor stepper 4 fasa ini memiliki rasio gigi 1 banding 64 dan 32 *teeth* dimana ada 8 *teeth* di tiap fasanya, tiap *teeth* berjarak  $11,25^\circ$  yang berarti tanpa *gear box* hanya membutuhkan 32 *step* untuk 1 putaran penuh pada porosnya. Namun untuk memperhalus dan meningkatkan presisi dari motor tersebut ditambah *gear box* dengan rasio 1 banding 64 yang berarti 1 putaran penuh dari poros *gear box* membutuhkan 64 putaran pada poros motor, sedangkan jumlah *step* untuk memutar 1 putaran penuh poros *gear box* adalah 32 *step* ( $1 \text{ putaran penuh poros motor} \times 64$  (64 putaran untuk memutar 1 putaran penuh poros *gear box*) = 2048 *step*).

Dengan adanya *gear box* maka *stride angle* yang awalnya  $11,25^\circ$  berubah menjadi  $0,18^\circ$ , hal ini karena  $360^\circ$  (1 putaran penuh)  $\div$  2048 *step* =  $0,18^\circ$  yang artinya satu kali *step* atau satu sinyal kotak akan menggerakkan porosnya sebanyak  $0,18^\circ$ . Ini berlaku hanya untuk *driving mode Full Step* dan *Wave Step*, dan dengan DC resistansi 50 Ohm, maka dapat ditentukan maksimal arus yang mengalir pada motor ini adalah 250 mA.

#### 4.2.4. Kebutuhan Perangkat Lunak

Perangkat lunak dibutuhkan pula untuk mengatur kontroler unit agar sesuai dengan tujuan sistem ini dibangun, maka dari itu untuk mencapai hal tersebut perangkat lunak yang dibutuhkan adalah

1. *Bootloader* Arduino Nano untuk digunakan pada kontroler unit ATmega328P.
2. Arduino IDE versi 1.8.5 Desktop untuk membuat program pada kontroler unit yang akan menjadi penentu perilaku sistem.
3. *Library* Wire.h untuk dapat menggunakan komunikasi *Serial* I2C yang berguna untuk berkomunikasi dengan sensor *IMU*.





## BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan membahas tentang Perancangan dan Implementasi dari sistem yang dibangun diantaranya adalah perancangan sub sistem perangkat keras *Power Supply Unit*, perancangan sistem perangkat keras secara keseluruhan, dan perancangan sistem perangkat lunak serta implementasi sub sistem perangkat keras *Power Supply Unit*, implementasi sistem perangkat keras secara keseluruhan, dan implementasi perangkat lunak.

### 5.1. Perancangan

Pada sub bab ini akan menjabarkan mengenai perancangan saja yang terdiri dari dua buah perancangan yaitu

1. Perancangan Perangkat Keras Elektronika
2. Perancangan Perangkat Lunak

Pada poin nomor 1 tentang perancangan perangkat keras elektronika meliputi perancangan bagian *power supply unit*, *driver* dan kontroler unit, sensor unit serta bagaimana mekanisme sistem dapat mengatur ketinggian pancaran cahaya dan poin nomor 2 membahas tentang perancangan perangkat lunak seperti algoritma yang digunakan, *library* yang digunakan, serta perangkat lunak lain yang mendukung.

#### 5.1.1. Perancangan Perangkat Keras: *Power Supply Unit*

Perancangan *Power Supply Unit* dilakukan dengan beberapa tahap yaitu menentukan masukan sumber daya listrik, menentukan kebutuhan tegangan yang diperlukan oleh setiap sub sistem atau menentukan keluaran yang dibutuhkan, dan menentukan cara untuk membuat masukan sumber daya listrik menjadi keluaran yang dibutuhkan.

1. Menentukan masukan sumber daya listrik.

Sumber daya listrik untuk kendaraan dipastikan menggunakan baterai yang berarti ber arus searah (*Direct Current*), baterai kendaraan untuk roda dua dan roda empat memiliki tegangan sebesar 12 Volt.

2. Menentukan keluaran sumber daya listrik yang dibutuhkan.

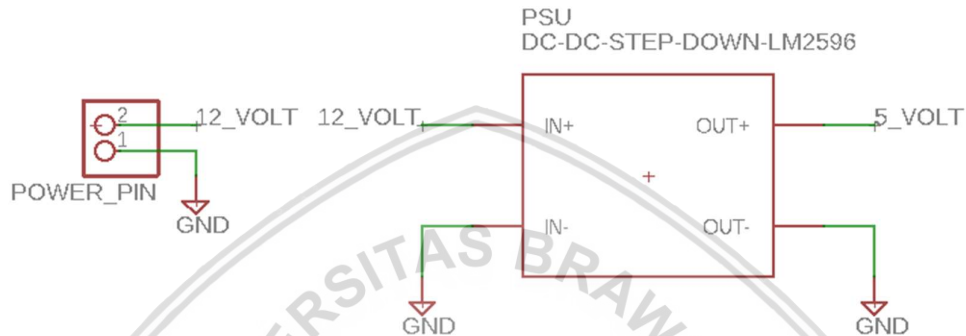
Keluaran sumber daya listrik yang dibutuhkan bergantung dari perangkat keras yang digunakan oleh sistem, pada penelitian ini perangkat keras sistem yang berupa kontroler unit, sensor IMU, dan aktuator stepper motor memerlukan tegangan DC sebesar 5 Volt.

3. Menentukan cara untuk membuat masukan sumber daya listrik menjadi keluaran sumber daya listrik yang dibutuhkan.



Berdasarkan poin nomor 2, sumber daya listrik keluaran harus bertegangan 5 Volt. Maka dari itu dibutuhkan penurun tegangan dari 12 Volt ke 5 Volt.

Hasil dari ketiga poin diatas dapat disimpulkan bahwa butuh penurun tegangan dari 12 Volt ke 5 Volt, ada banyak IC maupun modul untuk menurunkan tegangan yang dapat digunakan namun karena tegangan yang diturunkan banyak dan arus melebihi dari 100 mA maka penulis memutuskan untuk menggunakan penurun tegangan tipe *Switching* atau biasa dikenal dengan *Buck Converter*. Gambar 5.1 menggambarkan penggunaan *Buck Converter*.



**Gambar 5.1 Power Supply Unit**

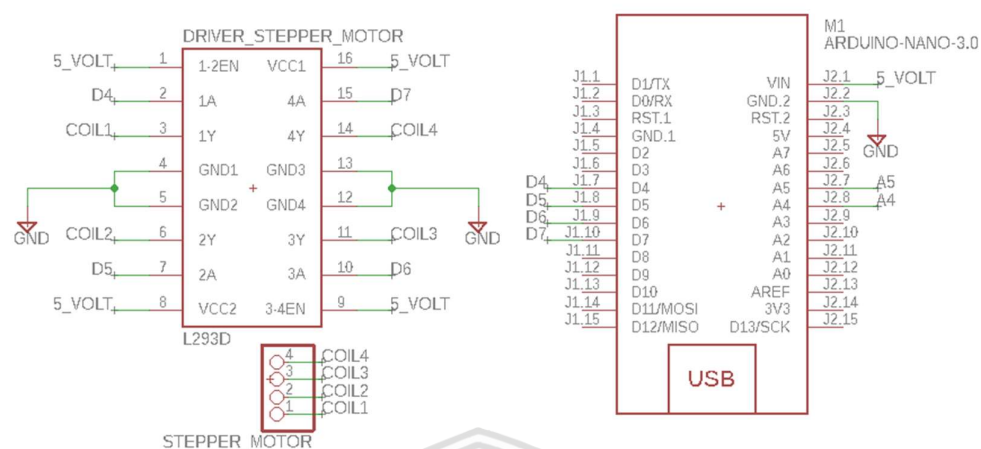
Pada bagian *rail* atas sebelah kiri dari modul merupakan tegangan dari baterai kendaraan yang nilainya 12 Volt yang terhubung langsung dengan PSU pada pin IN+, hasil regulasi tegangan dari modul ini berada pada *rail* atas sebelah kanan dari modul yaitu pin OUT+ yang dapat dengan aman digunakan pada sistem yaitu dengan nilai tegangan 5 Volt. Sedangkan untuk tiga pin lain yaitu pin 1 pada POWER\_PIN, IN- dan OUT- pada PSU terhubung pada GND atau kutub negatif baterai kendaraan.

### 5.1.2. Perancangan Perangkat Keras: *Driver*, Kontroler Unit dan Aktuator

*Driver* yang berguna untuk membantu kontroler unit dalam mengontrol aktuator, selain membantu kontroler unit hal ini juga mempermudah pengendalian aktuator serta memberikan perlindungan ekstra pada kontroler unit terhadap aktuator yang memiliki rangkaian induksi. Tahap – tahap merancang sub sistem aktuator adalah

1. Menentukan sinyal kontrol dari kontroler unit
2. Menentukan perangkat tambahan atau *Driver* yang dapat bekerja sesuai dengan sinyal kontrol dari kontroler

Sinyal kontrol dari kontroler unit adalah sinyal kotak untuk mengaktifkan tiap koil dari stepper motor, tegangan yang diperlukan adalah 5 Volt serta arus 250 mA. Dari berbagai *driver* yang tersedia dipasaran L293D adalah yang paling cukup dan tidak berlebihan baik dari segi harga maupun segi spesifikasi.



**Gambar 5.2 Driver, Kontroler Unit dan Aktuator**

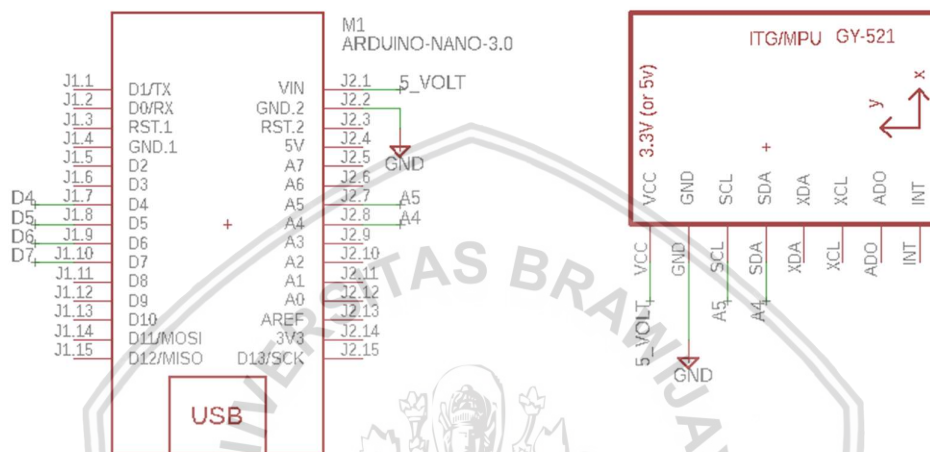
Gambar 5.2 menggambarkan koneksi untuk *driver* L293D Driver Stepper Motor, Pin dari Stepper Motor itu sendiri serta Kontroler Unit. Tabel 5.1 menjelaskan koneksi antar pin dari L293D Driver, Kontroler Unit, dan Aktuator.

**Tabel 5.1 Koneksi Antar Pin Driver, Kontroler Unit dan Aktuator**

L293D Driver Stepper Motor Pin	Kontroler Unit Pin	Aktuator
1, 8, 9, dan 16	VIN / 5_VOLT (PSU)	-
4, 5, 12, dan 13	GND	-
2 / 1A	Digital Pin 4 (D4)	-
3 / 1Y	-	COIL1
6 / 2Y	-	COIL2
7 / 2A	Digital Pin 5 (D5)	-
10 / 3A	Digital Pin 6 (D6)	-
11 / 3Y	-	COIL3
14 / 4Y	-	COIL4
15 / 4A	Digital Pin 7 (D7)	-

### 5.1.3. Perancangan Perangkat Keras: Sensor dan Kontroler Unit

Salah satu yang utama dari sistem ini adalah sensor inersia, pada sub bab ini akan dijelaskan perancangan sensor inersia tersebut. Sensor inersia yang tersedia dipasaran umumnya adalah MPU6050 6 *degree of freedom* yang terdiri dari *accelerometer* dan *gyroscope* dalam satu *broad*. Sensor tersebut menjadi satu dengan komponen – komponen lain dalam *breakout board* GY-521. Gambar 5.3 akan menggambarkan koneksi antara sensor dengan kontroler unit.



Gambar 5.3 Koneksi Sensor dan Kontroler Unit

Sensor IMU GY-521 berada pada sisi sebelah kanan dari gambar, ada 4 buah pin yang digunakan oleh sensor tersebut

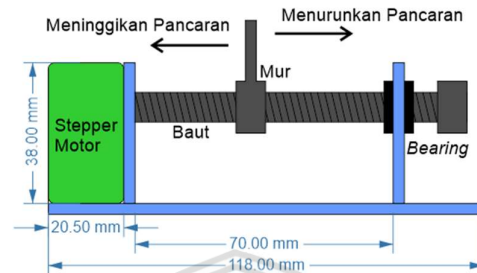
Tabel 5.2 Koneksi Antar Pin Sensor dan Kontroler Unit

Sensor GY-521	Kontroler Unit Pin
VCC	VIN / 5_VOLT (PSU)
GND	GND
SDA	Analog Pin 4 (A4)
SCL	Analog Pin 5 (A5)

Untuk Sensor ini bisa digunakan hingga 2 buah dengan memanfaatkan Pin ADO dari sensor yang merupakan pilihan dari alamat perangkat sensor tersebut, pada Gambar 5.5 pin ADO dibiarkan tak terkoneksi karena pin tersebut telah diberi *internal pull down* resistor yang menjadikannya bernilai 0 atau LOW. Kontroler Unit sendiri pun membutuhkan listrik juga untuk beroperasi, maka dari itu pin Vin dan GND dihubungkan pada *Rail* 5 Volt dan GND.

#### 5.1.4. Perancangan Perangkat Keras: Mekanisme Aktuasi

Bagian ini menjelaskan bagaimana sistem dapat bekerja mengatur ketinggian pancaran cahaya lampu. Gambar 5.4 menggambarkan rancangan mekanisme aktuasi.

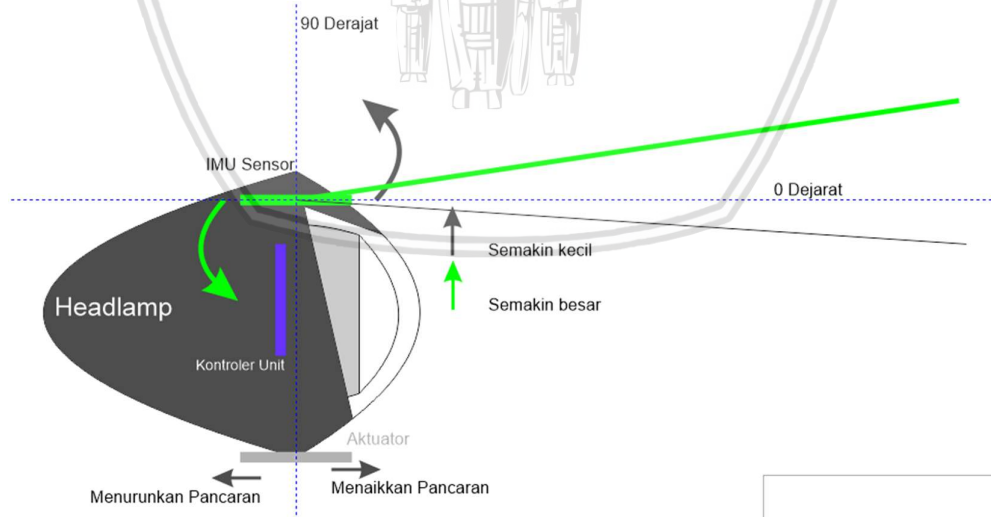


**Gambar 5.4 Mekanisme Aktuasi**

Pada Gambar 5.4 terdapat beberapa bagian yaitu Stepper Motor (Hijau), Mur dan Baut (Abu – Abu), serta *Bearing* (Hitam). Semua bagian itu ditopang oleh wadah akrilik setebal 3 mm (Biru) yang telah dibentuk sedemikian rupa sehingga pas. Bekerjanya sistem ini adalah dengan memutarnya Stepper Motor yang berakibat pada maju / mundurnya baut yang dikaitkan dengan *headlamp*.

#### 5.1.5. Perancangan Perangkat Keras: Peletakan Komponen Sistem

Bagian terakhir dari Perancangan Perangkat Keras, setelah semua dirancang maka terakhir adalah peletakan dari semua komponen agar sistem dapat bekerja. Gambar 5.5 menggambarkan peletakan dari semua perangkat keras sistem :



**Gambar 5.5 Peletakan Perangkat Keras Sistem**

Sensor (Hijau) diletakkan pada bagian atas *headlamp* karena engsel dari *headlamp* berada dilokasi tersebut sehingga akan menggerakkan sensor saat ada perubahan sudut baik efek dari adanya beban tambahan pada sepeda motor

maupun efek dari aktuator. Peletakan sensor pada *headlamp* ini terbalik secara sudut *Yaw*, hal ini dikarenakan kabel sensor akan membuat sensor tidak menempel dengan baik pada *headlamp* jika arahnya benar. Efek dari terbaliknya penempatan sensor ini adalah terbaliknya juga pembacaan sensor terhadap sudut dari *headlamp* jika dihitung menggunakan rumus.

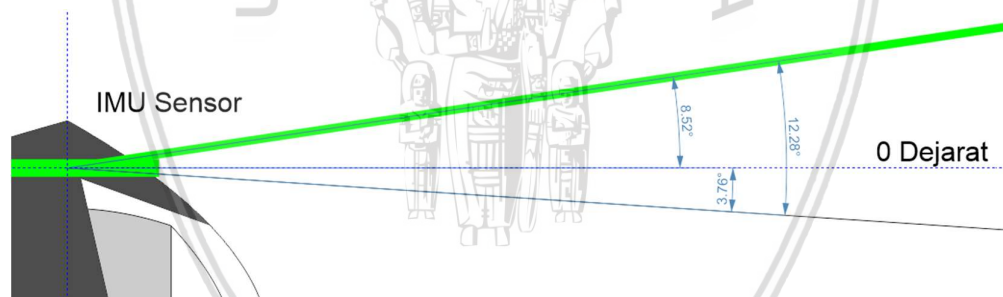
Kontroler Unit berada dibawah sensor, tempat ini dipilih karena selain dekat dengan sensor dekat juga dengan aktuator sehingga tidak memerlukan kabel terlalu panjang dan tempat ini juga sedikit lebih aman dari air karena berada didalam *headlamp*.

Aktuator berada dibagian bawah *headlamp*, hal ini karena pengaturan ketinggian pancaran cahaya dari pabrik berada pada bagian bawah.

Seperti yang terlihat pada Gambar 5.5 garis dan panah hijau, jika *headlamp* naik yang berarti pancaran cahaya juga naik dan sudut *headlamp* mengecil karena mendekati 0 derajat. Namun bacaan sensor justru membesar karena arah rotasinya menjauhi 0 derajat sehingga ada jarak yang tetap antara sudut *headlamp* dan sudut bacaan sensor. Jarak antar sudut tersebut bisa diperoleh dengan Persamaan 5.1 berikut.

$$\text{Jarak Antar Sudut} = \text{Sudut Bacaan Sensor} + \text{Sudut Headlamp} \quad (5.1)$$

$$\text{Sudut Bacaan Sensor} = \text{Jarak Antar Sudut} - \text{Sudut Headlamp} \quad (5.2)$$



Gambar 5.6 Contoh Jarak Antar Sudut

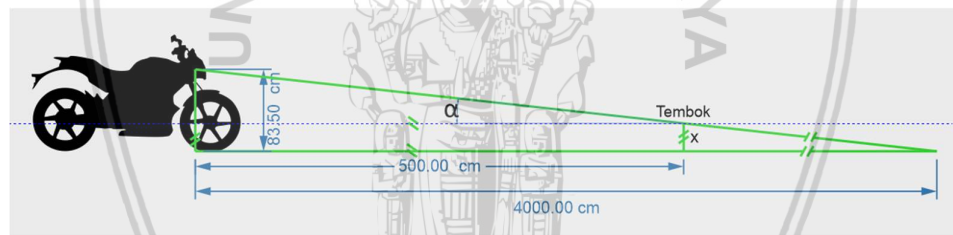
Contoh untuk mengukur jarak antar sudut dapat dilihat pada Gambar 5.6 dimana garis hitam dibagian bawah adalah garis pancaran cahaya dari *headlamp*, garis hijau dibagian atas adalah garis dari arah bacaan sensor, dan garis putus – putus pada horisontal adalah garis rata 0 derajat. Pada contoh Gambar 5.6 sudut *headlamp* adalah 3.76° dan sudut bacaan sensor adalah 8.52° maka jarak antar sudutnya adalah 12.28°. Masalah ini akan berpengaruh pada set poin karena sudut yang dijadikan set poin adalah sudut *headlamp*, maka dari itu sudut *headlamp* perlu di jadikan sudut bacaan sensor.

Untuk merubah sudut *headlamp* menjadi sudut bacaan sensor diperlukan data – data berupa :

1. Panjang Pancaran Cahaya (cm)
2. Tinggi Sumber Cahaya (cm)
3. Sudut Bacaan Sensor (derajat)
4. Sudut *Headlamp* (derajat)

Setelah data terkumpul maka dapat dicari jarak antar sudutnya dengan Persamaan 5.1, kemudian menentukan sudut *headlamp* berdasarkan peraturan pemerintah dilanjutkan dengan mengubahnya menjadi sudut bacaan sensor. Hal ini akan menjadi set poin untuk sistem.

Skenario untuk mendapatkan data diatas adalah dengan melakukan percobaan dengan jarak dan tinggi pancaran cahaya yang telah ditentukan, dalam hal ini pancaran cahaya seharusnya adalah 40 meter diperpendek menjadi 500 cm dengan batas tembok agar mempermudah percobaan. Dengan adanya tembok sebagai batas maka akan menambah 1 poin lagi yaitu tinggi pancaran cahaya pada tembok (x) sebagai ganti panjang pancaran cahaya seharusnya yang panjangnya mencapai 40 meter, Gambar 5.7 menggambarkan skenario percobaan.



**Gambar 5.7 Skenario Percobaan**

Saat ini ada beberapa hal yang telah diketahui berdasarkan Gambar 5.7 diatas

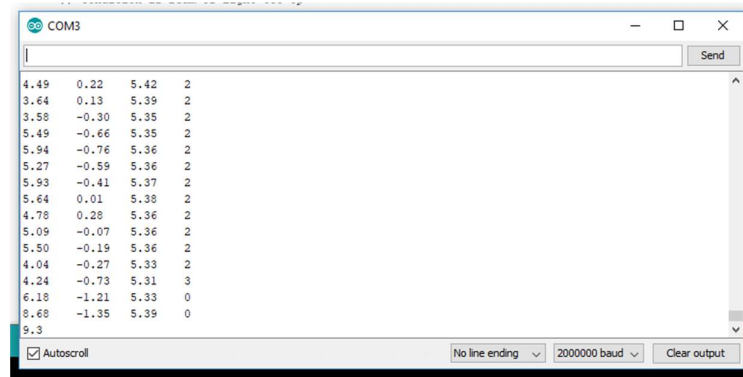
1. Panjang Pancaran Cahaya = 500 cm
2. Tinggi Sumber Cahaya = 83,5 cm

Serta ada 3 hal yang perlu dicari yaitu

1. Nilai Sudut Bacaan Sensor
2. Tinggi Pancaran Cahaya pada Tembok
3. Nilai Sudut *Headlamp* berdasarkan Tinggi Sumber Cahaya, Tinggi Pancaran pada Tembok, serta Panjang Pancaran Cahaya



Hasil dari nilai sudut bacaan sensor dapat dilihat pada Gambar 5.8 berikut



4.49	0.22	5.42	2
3.64	0.13	5.39	2
3.58	-0.30	5.35	2
5.49	-0.66	5.35	2
5.94	-0.76	5.36	2
5.27	-0.59	5.36	2
5.93	-0.41	5.37	2
5.64	0.01	5.38	2
4.78	0.28	5.36	2
5.09	-0.07	5.36	2
5.50	-0.19	5.36	2
4.04	-0.27	5.33	2
4.24	-0.73	5.31	3
6.18	-1.21	5.33	0
8.68	-1.35	5.39	0
9.3			

**Gambar 5.8 Nilai Sudut Bacaan Sensor**

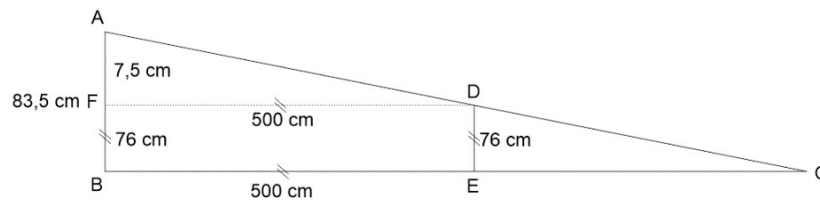
Data ketiga dari kiri adalah data untuk satuan derajat, dimana rata – rata dari nilai tersebut adalah  $5,36^{\circ}$ .

Tinggi pancaran pada tembok dapat dilihat pada Gambar 5.9 berikut



**Gambar 5.9 Tinggi Pancaran Cahaya pada Tembok**

Pada Gambar 5.9 tinggi pancaran cahaya pada tembok adalah 76 cm, hal ini telah menjawab poin nomor 2. Terakhir adalah mencari sudut *headlamp*, Gambar 5.10 berikut menjelaskan cara mendapatkan sudut *headlamp*.



**Gambar 5.10 Mencari Sudut Headlamp /  $\angle ADF$**

Dengan *Arc Tan* maka besar  $\angle ADF$  yang mewakili sudut *headlamp* dapat dicari

$$\angle ADF = \tan^{-1} \left( \frac{7.5 \text{ cm}}{500 \text{ cm}} \right) \quad (5.3)$$

$$\angle ADF = \tan^{-1}(0.015) \quad (5.4)$$

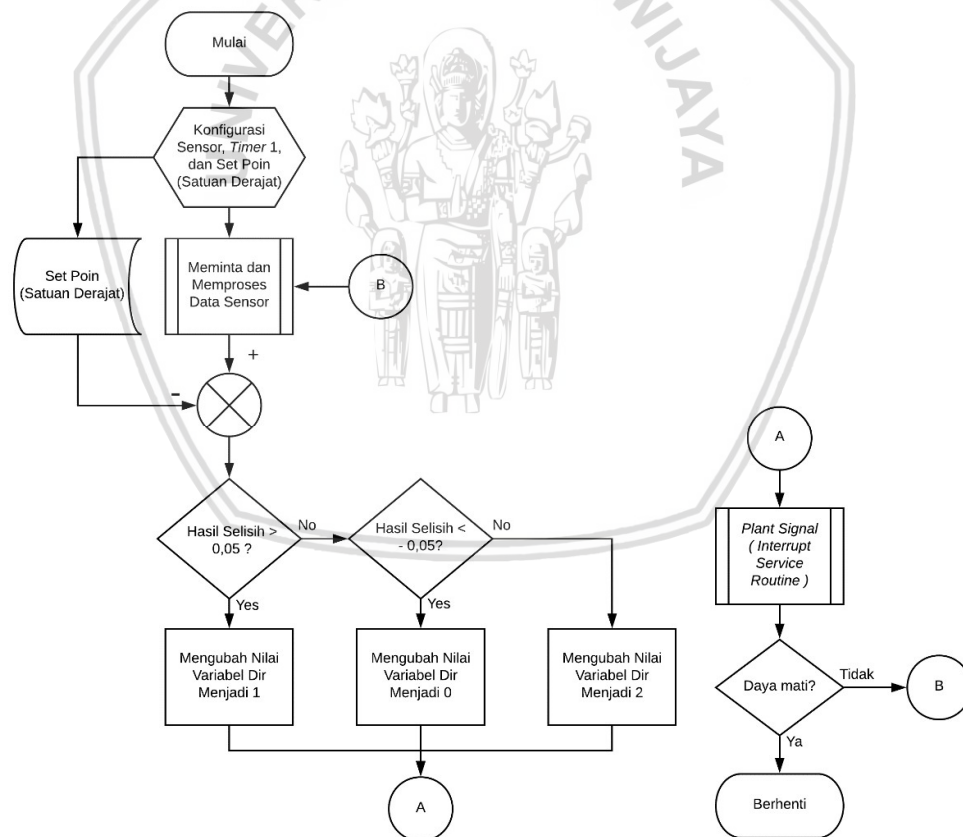
$$\angle ADF = 0,8593^\circ \quad (5.5)$$

$$\angle ADF \approx 0,86^\circ \quad (5.6)$$

Jika sudut bacaan sensor sebesar  $5,36^\circ$  dan sudut *headlamp* sebesar  $0,86^\circ$  maka jarak antar sudut sesuai dengan Persamaan 5.1 adalah  $5,36^\circ + 0,86^\circ = 6,22^\circ$ . Dari hasil ini kita dapat mencari sudut bacaan sensor yang nantinya akan dijadikan set poin dengan cara mengurangi jarak antar sudut ( $6,22^\circ$ ) dengan sudut *headlamp* yang telah dikalkulasi berdasarkan peraturan pemerintah.

### 5.1.6. Perancangan Perangkat Lunak

Perangkat lunak berisi instruksi – instruksi yang akan dikerjakan oleh sistem secara otomatis, Gambar 5.13 berikut adalah alur instruksi dari program sistem.



Gambar 5.11 Alur Instruksi Sistem

## 1. Konfigurasi *Sensor*, *Timer 1*, dan Set Poin

Dalam bagian ini memuat 3 buah pembahasan yaitu Konfigurasi *Sensor*, Konfigurasi *Timer 0*, dan Konfigurasi Set Poin, masing – masing akan dibahas pada sub dibawah ini.

### 1.1. Konfigurasi *Sensor*

Konfigurasi sensor dilakukan untuk pertama kali sensor tersebut digunakan, ada beberapa hal yang perlu diatur agar sensor dapat bekerja seperti yang diharapkan diantaranya ada menonaktifkan *Sleep Mode* pada sensor, mengatur *Full Scale Range* dan *Sensitivity Scale Faktor* dari *Accelerometer* dan *Gyroscope*.

#### 1.1.1. Register *PWR\_MGMT\_1* / *Power Management 1*

Menurut register *map* dari MPU6050 untuk menonaktifkan *Sleep Mode* maka harus mengakses register *PWR\_MGMT\_1* atau *Power Management 1* yang beralamat di 0x6B kemudian menghapus nilai pada bit ke 6 yang merupakan bit untuk mengatur *Sleep Mode*. Gambar 5.12 berikut akan menggambarkan isi dari register *PWR\_MGMT\_1*.

4.28 Register 107 – Power Management 1 PWR_MGMT_1									
Type: Read/Write									
Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6B	107	DEVICE_RESET	SLEEP	CYCLE	-	TEMP_DIS	CLKSEL[2:0]		

**Gambar 5.12 Register *PWR\_MGMT\_1***

Pada register *PWR\_MGMT\_1* ada 4 hal lain yang dapat diatur selain untuk mengatur *Sleep Mode*, diantaranya adalah

#### a. *CLKSEL[2:0]*

Pada bit 0 sampai 2 digunakan untuk memilih sumber dari *clock* sensor, MPU6050 telah tersedia *internal oscillator* sebesar 8 MHz. Maka dari itu peneliti memilih untuk menggunakan sumber tersebut, caranya adalah dengan menulis nilai 0 (nol) pada ketiga bit *CLKSEL* tersebut.

#### b. *TEMP\_DIS*

Pada bit 3 ini digunakan untuk mengatur temperatur sensor, karena letak register untuk mengakses data temperatur sensor berada di tengah – tengah antara *Accelerometer* dan *Gyroscope* maka untuk mempermudah dalam kode program pengaksesan data *Accelerometer* dan *Gyroscope* peneliti memutuskan untuk mengaktifkan temperatur sensor dengan cara menulis nilai 0 (nol) pada bit 3.

c. *CYCLE*

*CYCLE* adalah bit yang berfungsi untuk membangunkan sensor dalam rangka mengambil data *Accelerometer* dan *Gyroscope* dalam periodik waktu tertentu selama sensor dalam kondisi *Sleep Mode*, karena sensor tidak dalam kondisi *Sleep Mode* maka peneliti memutuskan untuk menonaktifkan fungsi *CYCLE* ini dengan cara menulis nilai 0 (nol) pada bit 6.

d. *DEVICE\_RESET*

Bit 7 ini adalah bit untuk mengatur ulang pengaturan sensor ke pengaturan pabrik, karena sensor tidak ingin di reset maka bit ini harus ditulis dengan nilai 0 (nol).

Semua bit pada register *PWR\_MGMT\_1* telah ditentukan nilainya dari bit 0 sampai 7 berisi 0 (nol), maka dari itu nilai yang harus dituliskan pada register *PWR\_MGMT\_1* adalah *0b00000000* dalam biner atau *0x00* dalam heksa.

1.1.2. Register *GYRO\_CONFIG* / *Gyroscope Configuration*

Kemudian dilanjutkan dengan konfigurasi *Gyroscope*, kali ini hanya memilih *Full Scale Range* yang diinginkan dan tentunya yang sesuai dengan ruang lingkup kerja sistem. *Full Scale Range* yang tersedia ada 4 pilihan yaitu *FS\_SEL* 0 sampai 3, dengan rentang bacaan dari maksimal :

- $\pm 250^{\circ}/s$  untuk *FS\_SEL* 0 dengan *Sensitivity Scale Factor* 131 LSB/( $^{\circ}/s$ )
- $\pm 500^{\circ}/s$  untuk *FS\_SEL* 1 dengan *Sensitivity Scale Factor* 65,5 LSB/( $^{\circ}/s$ )
- $\pm 1000^{\circ}/s$  untuk *FS\_SEL* 2 dengan *Sensitivity Scale Factor* 32,8 LSB/( $^{\circ}/s$ )
- $\pm 2000^{\circ}/s$  untuk *FS\_SEL* 3 dengan *Sensitivity Scale Factor* 16,4 LSB/( $^{\circ}/s$ )

*FS\_SEL* 0 adalah yang paling sensitif terhadap perubahan sudut dimana hasil dari bacaan *Gyroscope* nantinya akan dibagi dengan *Sensitivity Scale Factor*, namun dengan bertambahnya sensitifitas maka batasan percepatan sudut juga berkurang.

Dengan keadaan sistem pengatur ketinggian pancaran lampu ini yang tidak akan melebihi  $250^{\circ}/s$ , maka akan sangat tepat memilih skala yang sensitifitasnya paling tinggi. Untuk memilih *FS\_SEL* 0 pada konfigurasi *Gyroscope* maka dapat dilakukan dengan cara menulis ke 8 bit pada register *0x1B* dengan 0 (nol). Gambar 5.13 berikut akan menggambarkan isi dari register *GYRO\_CONFIG* / *Gyroscope Configuration* dan cara untuk memilih *FS\_SEL* 0.

#### 4.4 Register 27 – Gyroscope Configuration GYRO\_CONFIG

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1B	27	XG_ST	YG_ST	ZG_ST	FS_SEL[1:0]		-	-	-

**Gambar 5.13 Register GYRO\_CONFIG**

Pada bit 0 sampai 2 tidak digunakan maka akan bernilai 0 (nol), untuk memilih *FS\_SEL* 0 maka pada bit 3 dan 4 diisi dengan nilai biner dari 0 yaitu 0 (nol) dan 0 (nol) untuk masing – masing bit 3 dan 4. Selanjutnya untuk bit 5 sampai 7 yang berisi pengaturan untuk melakukan *Self Test* pada masing – masing sudut X, Y, dan Z sensor *Gyroscope*, karena saat ini tidak dibutuhkan *Self Test* maka ketiga bit tersebut diberi nilai 0 (nol).

Semua bit pada register *GYRO\_CONFIG* telah ditentukan nilainya dari bit 0 sampai 7 berisi 0 (nol), maka dari itu nilai yang harus dituliskan pada register *GYRO\_CONFIG* adalah *0b00000000* dalam biner atau *0x00* dalam heksa.

##### 1.1.3. Register ACCEL\_CONFIG / Accelerometer Configuration

Terakhir adalah konfigurasi *Accelerometer*, sama halnya dengan konfigurasi *Gyroscope* pada konfigurasi kali ini juga hanya memilih untuk menggunakan *Full Scale Range* pada *Accelerometer*. Ada beberapa pilihan *Full Scale Range* yang tersedia, *AFS\_SEL* 0 sampai 3 dengan rentang bacaan dari maksimal :

1.  $\pm 2\text{ g}$  untuk *AFS\_SEL* 0 dengan *Sensitivity Scale Factor* 16.384 LSB/g
2.  $\pm 4\text{ g}$  untuk *AFS\_SEL* 1 dengan *Sensitivity Scale Factor* 8.192 LSB/g
3.  $\pm 8\text{ g}$  untuk *AFS\_SEL* 2 dengan *Sensitivity Scale Factor* 4.096 LSB/g
4.  $\pm 16\text{ g}$  untuk *AFS\_SEL* 3 dengan *Sensitivity Scale Factor* 2.048 LSB/g

*AFS\_SEL* 0 adalah yang paling sensitif dimana nilai yang dihasilkan adalah 16.384 untuk tiap  $\pm 2\text{ g}$ , semakin besar hasilnya maka semakin banyak juga pergerakan yang dapat dideteksi oleh *Accelerometer* namun pergerakannya tidak boleh memiliki tekanan lebih dari  $\pm 2\text{ g}$ .

Sistem yang dibangun memiliki tekanan pergerakan dibawah  $\pm 2\text{ g}$  maka akan sangat tepat jika memilih *AFS\_SEL* 0. Untuk memilih *AFS\_SEL* 0 pada konfigurasi *Accelerometer* maka dapat dilakukan dengan cara menulis ke 8 bit pada register *0x1C* dengan 0 (nol). Gambar 5.14 berikut menggambarkan isi dari register *ACCEL\_CONFIG / Accelerometer Configuration* dan cara untuk memilih *AFS\_SEL* 0.



#### 4.5 Register 28 – Accelerometer Configuration ACCEL\_CONFIG

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]		-		

**Gambar 5.14 Register ACCEL\_CONFIG**

Pada bit 0 sampai 2 tidak digunakan maka akan bernilai 0 (nol), untuk memilih *AFS\_SEL* 0 maka pada bit 3 dan 4 diisi dengan nilai biner dari 0 yaitu 0 (nol) dan 0 (nol) untuk masing – masing bit 3 dan 4. Selanjutnya untuk bit 5 sampai 7 yang berisi pengaturan untuk melakukan *Self Test* pada masing – masing sudut X, Y, dan Z sensor *Accelerometer*, karena saat ini tidak dibutuhkan *Self Test* maka ketiga bit tersebut diberi nilai 0 (nol).

Semua bit pada register *GYRO\_CONFIG* telah ditentukan nilainya dari bit 0 sampai 7 berisi 0 (nol), maka dari itu nilai yang harus dituliskan pada register *ACCEL\_CONFIG* adalah *0b00000000* dalam biner atau *0x00* dalam heksa.

#### 1.2. Konfigurasi *Timer 1*

Selain sensor ada pula *timer 1* yang merupakan pengitung waktu berdasarkan *clock* dari CPU mikrokontroler, *timer 1* mempunyai bentang 16 bit. Pengaturan ini bertujuan untuk mengaktifkan *timer 1* yang secara *default* tidak aktif, mengatur *prescaler* serta *timer counting* untuk menentukan seberapa banyak *interrupt* yang dilakukan selama 1 detik. *Interrupt* ini berfungsi untuk mengirimkan sinyal aktuasi ke aktuator.

Berdasarkan *datasheet* mikrokontroler ATmega328P untuk mengaktifkan *timer 1* ada beberapa hal yang harus diatur terlebih dahulu, diantaranya adalah

- Mengatur mode operasi dari *timer 1* pada register *TCCR1A*
- Reset *prescaler timer 1* pada register *TCCR1B*
- Mengubah nilai *timer counter* pada register *TCNT1*
- Memilih *prescaler timer 1* pada register *TCCR1B*
- Mengaktifkan *timer 1* pada register *TIMSK1*

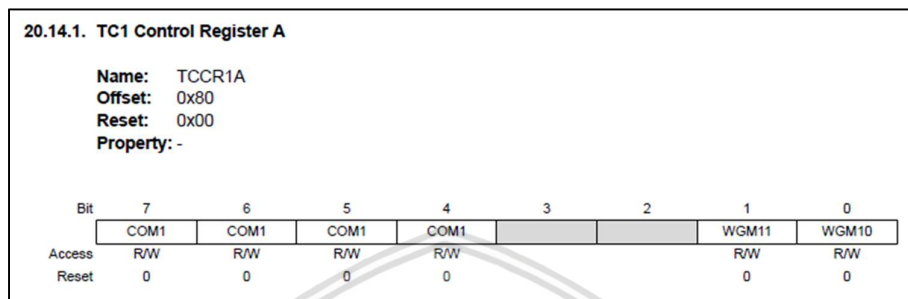
Setelah mengatur semuanya dan mengaktifkan *timer 1*, dibuatkan sebuah *Interrupt Service Routine* yang penjelasannya ada pada Gambar 5.20. *Interrupt Service Routine* akan dieksekusi tiap periodik tertentu setelah pengaturan *timer 1* ini selesai, selama *interrupti* dieksekusi maka program utama akan ditahan sementara sampai *interrupt* selesai dieksekusi. Hal ini membuat program berjalan selayaknya bersama – sama, namun sebenarnya satu – per satu.

Pada sub bagian dibawah ini akan menjelaskan tata cara mengatur *timer 1* sesuai dengan poin a sampai e.



### 1.2.1. Mengatur mode pada register *TCCR1A* / *Timer/Counter Control Register 1 A*

*TCCR1A* merupakan bagian dari *TCCR1B*, namun dalam *TCCR1A* lebih banyak mengatur tentang mode operasi dari *timer 1* yang berkaitan dengan pin *OC1A* dan *OC1B* dari mikrokontroler ATmega328P. Gambar 5.15 berikut adalah isi dari register *TCCR1A*.



**Gambar 5.15 Register *TCCR1A***

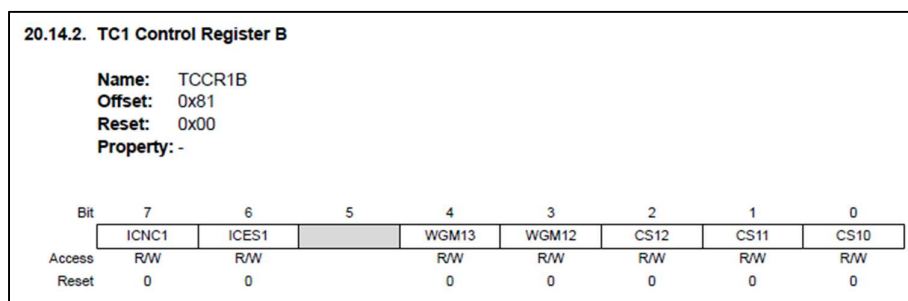
Dari bit 0 dan 1 ada *WGM10* dan *WGM11* untuk mengatur *Waveform Generation Mode*, bit ini untuk membuat gelombang pada pin fisik dari mikrokontroler seperti gelombang *Pulse Width Modulation (PWM)*. Karena sistem tidak membutuhkan gelombang tersebut maka untuk bit 0 dan 1 diberi nilai 0 (nol).

Bit 4 sampai 7 masing – masing adalah untuk mengatur pin *OC1A* untuk bit 4 dan 5, pin *OC1B* untuk bit 6 dan 7. Ada beberapa mode dalam pengaturan tersebut namun yang diperlukan sistem saat ini adalah tidak menghubungkan *timer 1* ini dengan pin fisik tersebut, hal ini dinamakan *normal mode*. Untuk mengatur ke *normal mode* maka bit 4 sampai 7 perlu diberi nilai 0 (nol).

Seluruh bit dari register *TCCR1A* telah ditentukan, semua bit bernilai 0 (nol). Maka dari itu register *TCCR1A* harus diisi dengan nilai 0b00000000 untuk biner atau 0x00 untuk heksa atau hanya 0 untuk desimal.

### 1.2.2. Reset *prescaler* pada Register *TCCR1B* / *Timer/Counter Control Register 1 B*

Register *TCCR1B* mempunyai pengaturan yang berbeda dengan *TCCR1A* namun ada bit yang merupakan gabungan dengan *TCCR1A*. Gambar 5.16 berikut akan menggambarkan isi dari register *TCCR1B*.



**Gambar 5.16 Register *TCCR1B***

Dari bit 0 sampai bit 2 berisi pengaturan untuk *CS10* sampai *CS12* yang merupakan kepanjangan dari *Clock Select*, ketiga bit ini digunakan untuk memilih *prescaler* yang berfungsi untuk membagi *clock* sebenarnya agar *interrupt* berjalan pada periode waktu tertentu.

Untuk saat ini perlu dilakukan reset pada *prescaler* karena pada bagian berikutnya akan mengganti nilai *timer counter*. Maka dari itu bit 0 sampai bit 2 diberi nilai 0 (nol).

Bit 3 dan 4 adalah bit untuk mengatur *Waveform Generation Mode* yang merupakan bagian dari bit *WGMxx* yang ada pada register *TCCR1A*, bit ini diberi nilai masing – masing 0 (nol) karena sistem tidak membutuhkan gelombang ini.

Bit 6 adalah bit *Input Capture Edge Select* atau *ICES1*, bit ini ada hubungannya dengan pin *ICP1* pada mikrokontroler. Pemberian nilai apapun pada bit ini tidak akan mempengaruhi *mode timer 1* jika bit *Input Capture Flag / ICF1* tidak diaktifkan pada register *TIFR1*, jadi untuk saat ini peneliti memutuskan untuk mengisinya dengan nilai 0 (nol).

Bit terakhir yaitu bit 7 yang mengatur tentang *Noise Canceled* pada pin *ICP1*, karena sistem tidak menggunakan pin tersebut maka bit ini dinonaktifkan dengan memberikan nilai 0 (nol).

Seluruh bit dari register *TCCR1B* telah ditentukan, semua bit bernilai 0 (nol). Maka dari itu register *TCCR1B* harus diisi dengan nilai 0b00000000 untuk biner atau 0x00 untuk heksa atau hanya 0 untuk desimal.

### 1.2.3. Mengubah nilai register *TCNT1 / Timer/Counter Counter 1*

Hal ini berhubungan dengan frekuensi dieksekusinya *interrupt* dalam sistem, hal - hal yang berhubungan dengan nilai register *TCNT1* adalah frekuensi *clock* dari *CPU* , *prescaler*, serta frekuensi eksekusi *interrupt* yang diinginkan. Berdasarkan Persamaan 5.7 dapat dicari nilai *TCNT1* agar frekuensi eksekusi *interrupt* sesuai dengan yang diinginkan.

Dalam hal ini penulis membuat *interrupt* tiap 3 milidetik atau 333,3333 Hz, hal ini adalah hasil dari percobaan dari perputaran stepper motor. 333, 3333 Hz menunjukkan perputaran paling cepat dari stepper motor, jika ditingkatkan lagi frekuensinya maka stepper motor tidak akan berputar karena koil motor tidak berenergi. Untuk mendapatkan frekuensi sebesar 333,333 hz, maka *prescaler* yang mencukupi adalah 8 *prescaler*.

$$TCNT1 = \left( 65536 - \left( \frac{freq\_CPU}{(prescaler \times freq\_target)} \right) \right) \quad (5.7)$$

Frekuensi *CPU* yang digunakan adalah 16.000.000 Hz, *prescaler* 8, dan frekuensi target adalah 333,3333 hz. Langkah – langkah mendapatkan nilai *TCNT1* dijelaskan pada Persamaan 5.8 sampai Persamaan 5.11

$$TCNT1 = \left( 65536 - \left( 16.000.000 / (8 \times 333,3333333) \right) \right) \quad (5.8)$$

$$TCNT1 = \left( 65536 - \left( 16.000.000 / 2.666,666667 \right) \right) \quad (5.9)$$

$$TCNT1 = (65536 - 6000) \quad (5.10)$$

$$TCNT1 = 59.536 \quad (5.11)$$

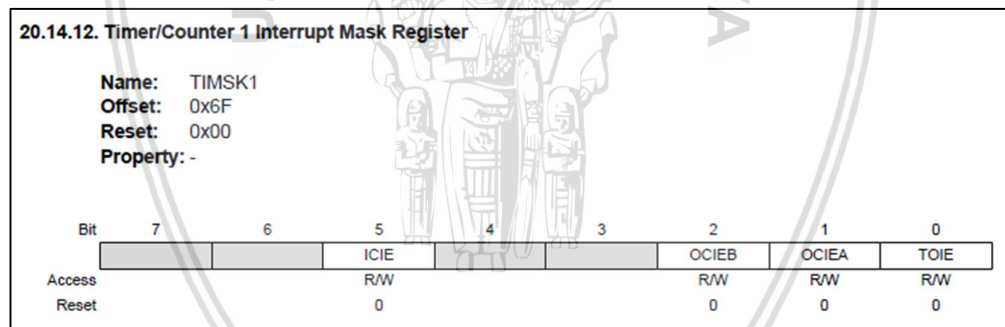
Dari Persamaan 5.11 nilai *TCNT1* agar *interrupt* dieksekusi setiap 3 milidetik adalah 59.536.

#### 1.2.4. Memilih *prescaler timer 1* pada register *TCCR1B*

Setelah melakukan reset pada register *TCCR1B*, kini dengan nilai *TCNT1* yang telah diatur maka *prescaler* dapat diatur sesuai dengan Persamaan 5.8 yaitu 8 *prescaler*. Cara untuk mengaturnya hanya perlu mengaktifkan atau menulis nilai pada bit 1 atau pada bit *CS11*.

#### 1.2.5. Mengaktifkan *timer 1* pada Register *TIMSK1* / *Timer/Counter 1 Interrupt Mask Register 1 B*

Register ini memiliki beberapa hal yang dapat diatur, Gambar 5.17 berikut akan menggambarkan isi dari register *TIMSK1*.



**Gambar 5.17 Register *TIMSK1***

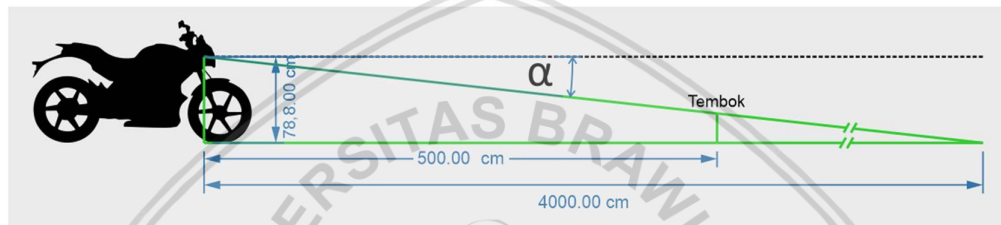
Bit 0 adalah *TOIE* atau *Overflow Interrupt Enable*, bit ini akan mengeksekusi *interrupt* setiap *TCNT1 Overflow* atau melebihi dari 65536. Karena sistem menggunakan mode ini maka bit ini diberi nilai 1 (satu) sedangkan untuk bit lainnya diberi nilai 0 (nol).

#### 1.3. Konfigurasi Set Poin

Inisialisasi set poin merupakan hal yang mutlak bagi kontroler, hal ini dikarenakan set poin menjadi acuan untuk menetapkan sistem yang dikontrol. Dalam menentukan set poin untuk sudut *Pitch* diperlukan referensi, referensi tersebut mengacu pada Peraturan Pemerintah Nomor 55 Tahun 2012 Pasal 24 Ayat 2 Tentang Kendaraan.

Menurut Peraturan Pemerintah Nomor 55 Tahun 2012 Pasal 24 Ayat 2 yang berbunyi “Untuk Sepeda Motor harus dilengkapi dengan lampu utama dekat dan lampu utama jauh paling banyak dua buah dan dapat memancarkan cahaya paling sedikit 40 (empat puluh) meter ke arah depan untuk lampu utama dekat dan 100 (seratus) meter ke arah depan untuk lampu utama jauh.” (Indonesia, 2012) menyatakan bahwa lampu utama dekat dapat memancarkan cahaya paling sedikit 40 meter ke arah depan dan 100 meter untuk lampu utama jauh.

Menanggapi hal tersebut maka untuk mendapatkan pancaran cahaya paling sedikit 40 meter ke arah depan maka harus ada sudut yang diatur oleh sistem, sudut tersebut adalah sudut *headlamp* /  $\alpha$  / *Pitch*. Dengan menggunakan teori trigonometri maka sudut *headlamp* /  $\alpha$  / *Pitch* dapat dicari, Gambar 5.18 berikut akan menggambarkan ilustrasi dari teori trigonometri yang akan digunakan :



**Gambar 5.18 Ilustrasi Pencarian Sudut *Pitch* ( $\alpha$ )**

Sudut yang akan dicari adalah sudut *headlamp* /  $\alpha$  / *pitch* pada Gambar 5.18, untuk mendapatkan sudut tersebut paling mudah menggunakan *Arc Tan* yang merupakan perbandingan antara tinggi peletakan *headlamp* dengan Jarak yang diinginkan.

Berikut adalah langkah – langkah untuk mencari set poin :

1. Ukur Tinggi Sumber Pancaran Cahaya
2. Cari Sudut *Headlamp*
3. Cari Sudut Bacaan Sensor dengan Persamaan 5.2

Tinggi sumber cahaya atau tinggi *headlamp* dari tanah adalah 78,8 cm, sedangkan untuk sudut *headlamp* yang berdasarkan tinggi 78,8 cm dan jarak pancaran yang diinginkan sejauh 40 m atau 4000 cm dapat dicari dengan Persamaan 5.12 sampai Persamaan 5.15 berikut :

$$\alpha = \tan^{-1} \left( \frac{78,8 \text{ cm}}{4000 \text{ cm}} \right) \quad (5.12)$$

$$\alpha = \tan^{-1}(0.0197) \quad (5.13)$$

$$\alpha = 1.128^\circ \quad (5.14)$$

$$\alpha \approx 1.2^\circ \quad (5.15)$$

Hasil pada Persamaan 5.14 dibulatkan menjadi  $1.2^\circ$  yang berarti sudut *headlamp* /  $\alpha$  / *pitch* semakin kebawah dan berakibat memendeknya jangkauan pancaran cahaya, Persamaan 5.16 sampai Persamaan 5.20 berikut adalah langkah

– Langkah mencari jangkauan pancaran cahaya yang baru ( $y$ ) dengan sudut *headlamp* /  $\alpha$  / *pitch* =  $1.2^\circ$  :

$$\tan 1.2 = \frac{78,8 \text{ cm}}{y} \quad (5.16)$$

$$0.020947014 = \frac{78,8 \text{ cm}}{y} \quad (5.17)$$

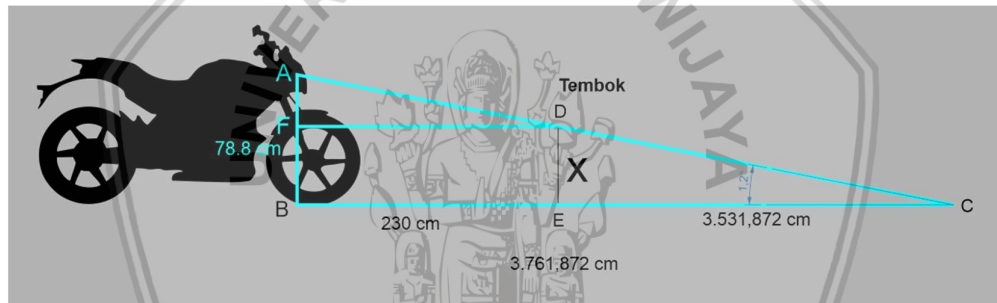
$$y = \frac{78,8 \text{ cm}}{0.020947014} \quad (5.18)$$

$$y = 3.761,87 \text{ cm} \quad (5.19)$$

$$y = 37,61 \text{ meter} \quad (5.20)$$

Dengan sudut *headlamp* /  $\alpha$  / *pitch* sebesar  $1.2^\circ$  maka jangkauan pancaran cahaya berkurang 2,39 meter, jangkauan sejauh 37,61 meter ini masih bisa ditoleransi.

Untuk tataletak dan jarak dari komponen pengujian seperti tinggi pancaran cahaya, jarak pancaran cahaya, dan lain sebagainya akan dijelaskan pada Gambar 5.19 berikut.



**Gambar 5.19 Tataletak dan Jarak Komponen Pengujian Sistem**

Pada Gambar 5.19 terdapat tembok (garis DE) tepat 230 cm didepan pancaran cahaya (titik A), peneliti melakukan ini agar mempermudah pengujian. Hanya dengan mengetahui panjang garis DE atau tinggi pancaran cahaya pada tembok maka nilai tersebut dapat mewakili jangkauan pancaran cahaya sebenarnya.

Hal – hal yang telah diketahui untuk dapat menentukan sistem bekerja sesuai perancangan atau tidak adalah

1. Jarak Sumber Cahaya ke Tembok (BE) = 230 cm
2. Tinggi Sumber Cahaya (AB) = 78,8 cm
3. Sudut *Pitch Headlamp* atau Set Poin ( $\angle ACB$ ) =  $1.2^\circ$
4. Jangkauan Cahaya Seharusnya (BC) = 3.761,872 cm



Hal – hal yang perlu diketahui untuk dapat menentukan sistem bekerja sesuai perancangan atau tidak adalah

1. Tinggi Pancaran Cahaya pada Tembok (DE) jika Sudut *Pitch Headlamp* =  $1,2^\circ$

Karena panjang garis DE menjadi acuan apakah sistem bekerja sesuai dengan perancangan atau tidak, maka perlu dicari panjang dari garis DE sesuai dengan sudut *pitch headlamp* yang ditentukan. Persamaan 5.21 sampai Persamaan 5.24 berikut akan menunjukkan berapa panjang garis DE seharusnya :

$$DE = \tan(1,2^\circ) \times 3.531,872 \text{ cm} \quad (5.21)$$

$$DE = 0,020947014 \times 3.531,872 \text{ cm} \quad (5.22)$$

$$DE = 73,98217191 \text{ cm} \quad (5.23)$$

$$DE \approx 74 \text{ cm} \quad (5.24)$$

Panjang garis DE seharusnya adalah 74 cm jika ingin jangkauan pancaran cahaya sejauh 37,61 meter, maka dari itu pada pengujian dengan tataletak komponen pengujian yang sama tinggi pancaran cahaya pada tembok harus setinggi 74 cm.

Selanjutnya adalah mencari sudut bacaan sensor yang nantinya akan dijadikan set poin oleh kontroler, menjadikan set poin berdasarkan dari bacaan sensor dilakukan agar kontroler unit tidak perlu menambah waktu untuk mengkalkulasi sudut *headlamp* /  $\alpha$  / *pitch* yang sebenarnya dari sudut bacaan sensor. Sudut bacaan sensor / set poin dapat dihitung dengan Persamaan 5.2, hasilnya ada pada Persamaan 5.27 berikut ini :

$$\text{Sudut Bacaan Sensor} = 6,22^\circ - 1,2^\circ \quad (5.25)$$

$$\text{Sudut Bacaan Sensor} = 5,02^\circ \quad (5.26)$$

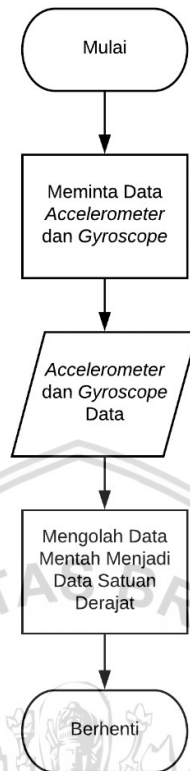
$$\text{Sudut Bacaan Sensor} \approx 5^\circ \quad (5.27)$$

Maka set poin adalah  $5^\circ$ .

2. Meminta, Menerima, dan Mengola Data Mentah

Pada *Predefined Process* pertama pada Gambar 5.13 adalah sebuah metode atau fungsi khusus untuk meminta, menerima, dan mengola data mentah atau *RAW Data* dari sensor yang masih berupa data percepatan perubahan sudut (*Angular Velocity*) untuk *Gyroscope* serta gaya gravitasi yang dipetakan berdasarkan sensitifitas tertentu untuk *Accelerometer*. Maka dari itu metode atau fungsi tersebut akan diperjelas oleh Gambar 5.20 pada halaman 41.





**Gambar 5.20 Predefined Process Meminta dan Memproses Data Sensor**

Pada Gambar 5.20 kontroler unit meminta data *Accelerometer* dan *Gyroscope*, dua data tersebut berada pada register yang berbeda sehingga harus diminta satu persatu, kemudian data tersebut diperoleh yang ditandai dengan simbol *flowchart* data, dan terakhir data tersebut diproses untuk menjadi satuan derajat.

#### 2.1. Meminta Data pada Sensor

Komunikasi yang digunakan oleh sensor adalah *Serial Inter-Integrated Circuit* atau *I2C*, dengan bantuan *library I2C* yang telah di sediakan oleh Nicholas Zambetti untuk *Arduino* maka dapat dengan mudah mengirim permintaan pada sensor. Langkah – langkahnya adalah sebagai berikut :

##### a. Memulai Koneksi dengan Sensor

Koneksi dengan sensor didahului dengan mengakses alamat dari perangkat sensor, pada perangkat kali ini alamat dari sensor adalah *0x68*.

##### b. Mengakses Register Tempat Data Sensor Tersimpan

Setelah mengakses alamat perangkat sensor, kemudian dilanjutkan dengan mengakses alamat register yang berisi data sensor. Ada 14 register dari ke 6 sudut (3 sudut *Accelerometer* dan 3 sudut *Gyroscope*) serta 1 register sensor suhu yang akan diakses oleh kontroler unit mulai dari alamat *0x3B* sampai *0x48*. Ada 14 register karena tiap register hanya menyimpan 8 bit data sedangkan panjang

datanya adalah 16 sehingga membutuhkan 2 register untuk menyimpan keseluruhan data.

c. Meminta Data Sensor

Bagian terakhir adalah meminta data dengan cara mengirimkan *request* dari alamat register pertama tempat data sensor disimpan, yaitu alamat register *0x3B*. Karena ada 14 register yang menyimpan data sensor untuk MPU6050 maka *request* yang diminta adalah sebanyak 14 register yang menampung data sensor dari alamat *0x3B* sampai *0x48*.

2.2. *Accelerometer* dan *Gyroscope* Data

Pada bagian ini adalah menerima data dari sensor dan menyimpannya pada variabel tertentu sesuai dengan data sensor.

2.3. Mengolah Data Sensor

Data yang diterima oleh kontroler unit adalah data mentah yang masih harus diproses dengan algoritma tertentu, dalam hal ini untuk memproses data *Accelerometer* menggunakan Pitagoras untuk mendapatkan nilai resultan vektor dari ke 3 sudut *Accelerometer* serta Trigonometri yaitu *arc sin* dengan parameter sudut *Y* dari sensor dibagi dengan resultan vektor yang telah di kalkulasi sebelumnya. Sedangkan untuk *Gyroscope* menggunakan Integral yang menjumlahkan hasil perkalian percepatan sudut (satuan *Gyroscope* derajat per detik) dengan lama waktu eksekusi program *loop*. Hasil dari keduanya kemudian digabungkan dengan *Complementary Filter*.

Dalam *Complementary Filter* terdapat sisi *High Pass Filter* yang digunakan oleh *Gyroscope* guna mem-filter frekuensi tinggi dari *Accelerometer* yang terdapat banyak *noise* dan menggunakan nilai frekuensi tinggi dari *Gyroscope* yang lebih stabil. Serta sisi *Low Pass Filter* yang digunakan oleh *Accelerometer* guna meloloskan nilai frekuensi rendah dari *Accelerometer* yang lebih stabil dalam jangka panjang dan mem-filter nilai frekuensi rendah dari *Gyroscope* yang tidak stabil dalam jangka panjang (terdapat istilah *drift* untuk *Gyroscope* yang berarti nilai bacaan sensor *Gyroscope* akan berubah seiring dengan bertambahnya waktu dikarenakan proses integral) (Debra, 2018). Berikut adalah rumus untuk menggunakan *Complementary Filter*.

$$angle = \alpha \times (angle + GyroD * dt) + (1 - \alpha) * (AccelA) \quad (5.28)$$

$$\alpha = \frac{t}{t+dt} \quad (5.29)$$

(Colton, 2007)

a. *Angle* adalah sudut hasil dari *Complementary Filter*.

- b. *Alpha* ( $\alpha$ ) adalah koefisien dari *filter* dimana nantinya akan menentukan porsi dari penggunaan sisi *Low Pass Filter* dan sisi *High Pass Filter*, nilai dari *Alpha* antara 0 dan 1.
- c. *Time Constant* ( $t$ ) adalah periode yang menjadi batasan penentu untuk lebih mengutamakan penggunaan sisi *Low Pass Filter* atau sisi *High Pass Filter*, nilai dari *Time Constant* harus lebih dari periode *noise Accelerometer* secara umum.
- d. *Delta Time* ( $dt$ ) adalah lamanya waktu eksekusi program *loop* atau periode sampel program *loop* yang dalam sistem ini ditentukan sebesar 2 milidetik atau 0,002 detik.
- e. *GyroD* adalah nilai mentah dari sensor yang telah di bagi dengan *Sensitivity Scale Factor* yang terdapat pada *datasheet* sensor.
- f. *AccelA* adalah nilai sudut dalam satuan derajat dari sensor *Accelerometer*.

Untuk menggunakan *Complementary Filter* terlebih dahulu mencari nilai dari *Time Constant* yang dapat di peroleh dengan mencari frekuensi *noise* dari ke tiga sudut *X*, *Y*, dan *Z* dari *Accelerometer*, lalu nilai *Time Constant* tersebut di gunakan untuk mencari nilai *Alpha*. Tabel 5.3 berikut menjelaskan rata-rata dari frekuensi *noise Accelerometer* yang akan di gunakan untuk mencari nilai *Time Constant*.

**Tabel 5.3 Noise Accelerometer**

Percobaan	Rata-rata Sudut X	Rata-rata Sudut Y	Rata-rata Sudut Z	Rata-Rata
1	90,18 Hz	113 Hz	91,775 Hz	98,32 Hz
2	93,42 Hz	111,89 Hz	82,19 Hz	95,83 Hz
3	95,38 Hz	129,71 Hz	93,87 Hz	106,32 Hz
Rata-Rata dari 3 kali percobaan				100,16 Hz

Nilai *noise* dari *Accelerometer* di dapat saat melakukan pembacaan terhadap data mentah dari tiap sudut dengan keadaan mesin sepeda motor menyala dan sepeda motor diam ditempat selama 2 detik dan menghasilkan 900 data. Kemudian di cari nilai puncak dari tiap data dan di cari juga periodenya, dari ke 900 data tersebut menghasilkan periode yang bermacam-macam sehingga di tentukan rata-ratanya lalu di ubah ke satuan Hertz.

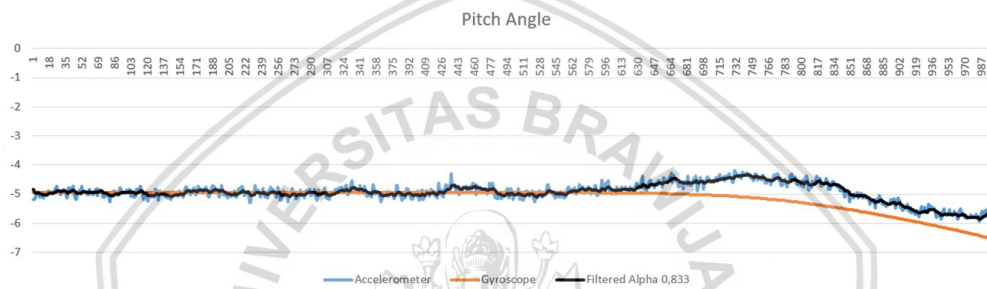
Dari hasil rata-rata ke 3 percobaan menghasilkan frekuensi *noise* sebesar 100,16 Hz yang jika di ubah ke periode menghasilkan waktu periode sebesar 0,00998 detik. Nilai tersebut adalah nilai *Time Constant* minimal yang di perbolehkan untuk mencari nilai *Alpha*.

Untuk mencari nilai *Alpha* dapat menggunakan Persamaan 5.29, langkah-langkahnya sebagai berikut.

$$\alpha = \frac{0,00998}{0,00998 + 0,002} \quad (5.30)$$

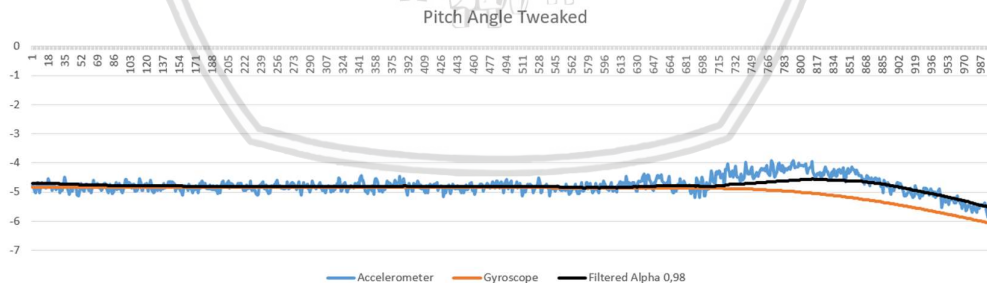
$$\alpha = 0,833 \quad (5.31)$$

Nilai *Alpha* sebesar 0,833 adalah nilai minimal yang dapat digunakan karena nilai *Alpha* berada diantara 0 dan 1 yang berarti nilai maksimal dari *Alpha* adalah 1. Dalam penggunaannya dalam *Complementary Filter* hal ini perlu dilakukan *tweaking* dari nilai *Alpha* 0,833 sampai 1,000 agar sinyal dihasilkan oleh *Complementary Filter* cukup bagus. Menurut hasil percobaan *Alpha* yang menghasilkan sinyal bagus adalah 0,98 dengan perbandingan sinyal sebagai berikut.



**Gambar 5.21 Sudut Pitch Alpha 0,833**

Pada Gambar 5.21 garis hitam yang menggambarkan sudut hasil dari *Complementary Filter* dengan *Alpha* 0,833 terlihat masih sedikit bergerigi mengikuti garis biru yang merupakan sudut dari *Accelerometer*. Bergerigi berarti sudut yang dihasilkan masih sedikit berubah-ubah meskipun tidak sebesar sudut dari *Accelerometer*.



**Gambar 5.22 Sudut Pitch Alpha 0,98**

Pada Gambar 5.22 garis hitam yang menggambarkan sudut hasil dari *Complementary Filter* dengan *Alpha* 0,98 terlihat lebih halus dari garis biru, dan tentunya juga garis hitam pada Gambar 5.22 lebih halus dari pada garis hitam pada Gambar 5.21. Garis yang halus berarti sudut yang dihasilkan sudah cukup stabil meskipun ada sedikit penundaan waktu saat ada perubahan sudut.

### 3. Summing Junction

Pada *Summing Junction* ini adalah inti dari *Close Loop System*, dimana set poin hasil perhitungan Persamaan 5.27 yang sebesar  $5^\circ$  diselisihkan dengan hasil bacaan sensor. Hasilnya akan memutuskan arah dari perputaran aktuator yang mana berefek pada naik atau turunnya pancaran cahaya.

#### 4. *Decision Point*

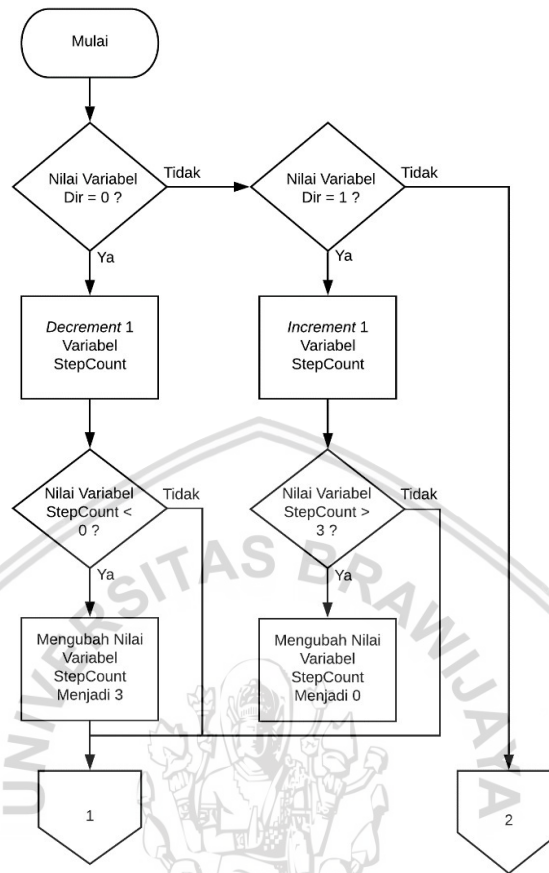
Pada bagian ini memberikan ruang pada sistem untuk mentoleransi error sebesar  $\pm 0,05^\circ$ . Hal ini akibat dari bacaan sensor yang selalu berubah – ubah pada ketelitian 2 angka dibelakang koma.

#### 5. Proses Mengubah Nilai Variabel Dir

Bergantung pada hasil dari *Summing Junction*, pada proses ini hanya mengubah nilai dari variabel Dir yang berakibat pada urutan *step* pada *interrupt service routine*. Hal ini berakibat pada arah perputaran stepper motor yang juga berakibat pada naik atau turunnya pancaran cahaya *headlamp*.

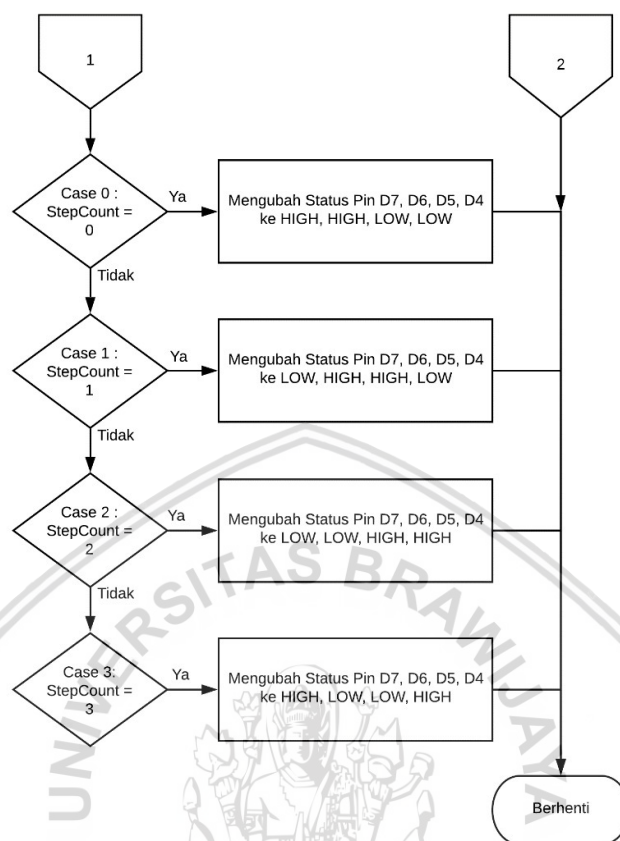
#### 6. *Interrupt Service Routine*

*Predefined Process* juga terdapat pada akhiran dari Gambar 5.13, pada proses ini adalah proses kontroler unit yang memberikan perintah pada aktuator berdasarkan hasil dari proses selisih pada metode *Close Loop System*. Proses tersebut akan diperjelas pada Gambar 5.21 berikut.



**Gambar 5.23 Predefined Process Plant Signal (Interrupt Service Routine)**





**Gambar 5.23 (lanjutan)**

Yang terjadi pada Gambar 5.21 diatas sebenarnya adalah proses ISR atau *Interrupt* yang dilakukan secara rutin tiap sekian milidetik, waktu rutin ini dieksekusi bergantung pada konfigurasi *Timer 1*. Dalam *sub routine* ini membaca nilai dari hasil selisih yang nilainya disimpan dalam variabel *Dir*, dari nilai variabel itu lah ditentukan step selanjutnya untuk aktuator. Tiap step akan berulang – ulang terus dengan penanda variabel *StepCount* yang bernilai antara 0 sampai 3 yang berarti ada 4 buah step yang berbeda, perbedaannya ada pada nilai dari variabel yang mengatur status Pin Digital pada kontroler unit.

#### 7. Decision Point

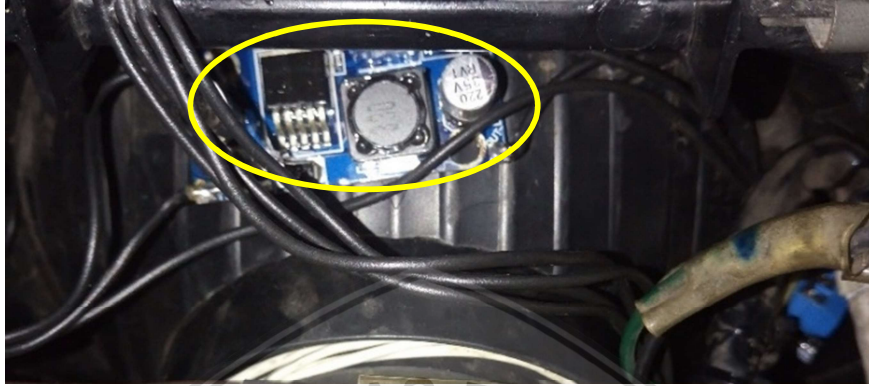
Bagian terakhir ini adalah pilihan apakah sistem harus berhenti atau terus mengulang program, maka dari itu selama ada daya listrik yang mengalir sistem tidak akan berhenti.

## 5.2. Implementasi

Pada sub bab ini akan membahas tentang pengimplementasian sesuai dengan sub bab 5.1 Perancangan.

### 5.2.1. Implementasi Perangkat Keras: *Power Supply Unit*

Implementasi dari sub bab 5.2.1 akan membahas tentang perangkat keras yang telah diimplementasikan. Gambar 5.22 menggambar perangkat keras *Power Supply Unit* yang telah diimplementasikan :

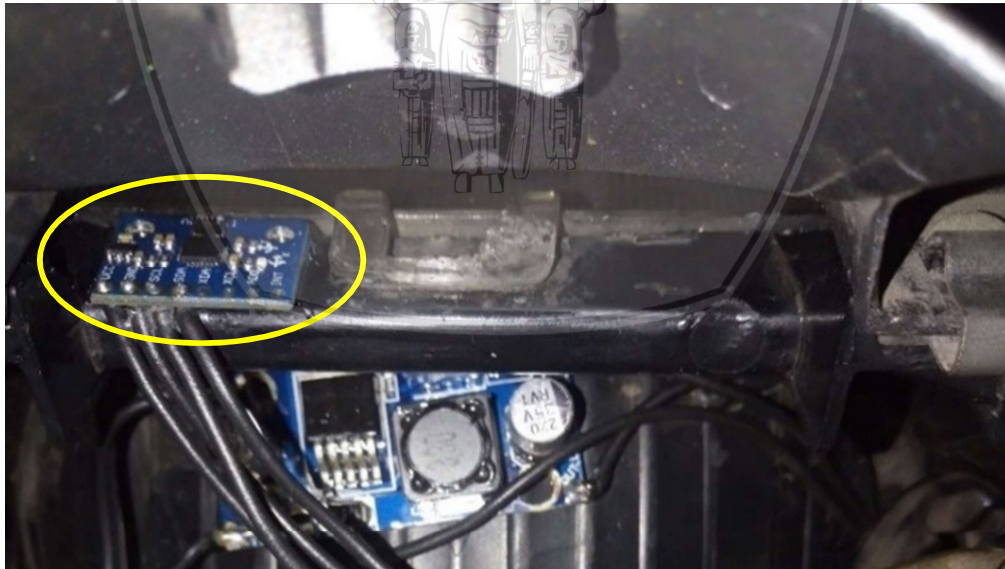


Gambar 5.24 Perangkat Keras : *Power Supply Unit*

Pada Gambar 5.22 terlihat modul berwarna biru adalah *Buck – Converter* yang menurunkan tegangan dari 12 Volt menjadi 5 Volt dengan arus maksimal 3 Amper.

### 5.2.2. Implementasi Perangkat Keras: Sensor

Sensor diletakkan pada bagian datar / horisontal pada *headlamp*, Gambar 5.23 menggambarkan peletakan dari sensor MPU6050 :

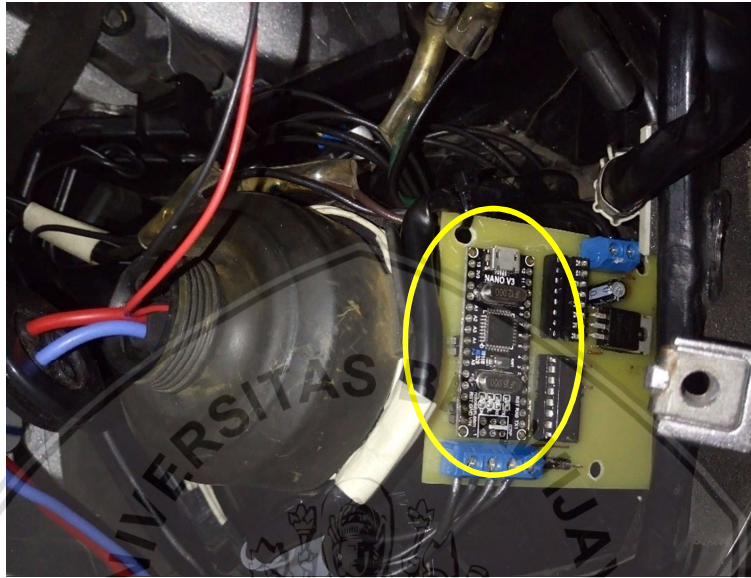


Gambar 5.25 Perangkat Keras : Sensor

Sensor berada dibagian kiri atas dari Gambar 5.23, peletakannya diatas dari PSU.

### 5.2.3. Implementasi Perangkat Keras: Kontroler Unit

Kontroler Unit adalah otak dari sistem, maka dari itu harus dapat menjangkau sensor maupun aktuator. Berdasarkan perancangan peletakan dari kontroler unit sesuai dengan Gambar 5.5, maka kontroler unit diletakkan pada bagian belakang *headlamp*. Gambar 5.24 menggambarkan peletakan kontroler Unit :



Gambar 5.26 Perangkat Keras : Kontroler Unit

### 5.2.4. Implementasi Perangkat Keras: Aktuator

Untuk aktuator harus dapat menggerakkan *headlamp*, pengatur gerakan *headlamp* berada pada bagian luar dari *headlamp*. Gambar 5.25 menggambarkan peletakan dari aktuator :



Gambar 5.27 Perangkat Keras : Aktuator

Karena berada dibagian luar, maka aktuator ditopang oleh *bracket* nomor polisi. *Bracket* tersebut dibuat sedemikian rupa hingga bisa menopang akuator serta nomor polisi itu sendiri.

### 5.2.5. Implementasi Perangkat Lunak

Pada sub bab ini akan membahas tentang perangkat lunak sesuai dengan perancangan perangkat lunak pada sub bab 5.1.6.

#### 1. Konfigurasi Sensor

Sensor perlu dikonfigurasi saat pertama kali digunakan. Tabel 5.3 berikut merupakan implementasinya pada perangkat lunak :

**Tabel 5.4 Konfigurasi Sensor**

Konfigurasi Sensor	
1	Wire.begin();
2	TWBR = 12;
3	Wire.beginTransmission(0x68);
4	Wire.write(0x6B);
5	Wire.write(0b00000000);
6	Wire.endTransmission();
7	Wire.beginTransmission(0x68);
8	Wire.write(0x1B);
9	Wire.write(0b00000000);
10	Wire.endTransmission();
11	Wire.beginTransmission(0x68);
12	Wire.write(0x1C);
13	Wire.write(0b00000000);
14	Wire.endTransmission();
15	Wire.beginTransmission(0x68);
16	Wire.write(0x1A);
17	Wire.write(0b00000011);
18	Wire.endTransmission();

#### 2. Konfigurasi *Timer 1*

Selain Sensor, sistem juga membutuhkan *timer 1* dari kontroler unit untuk mengeksekusi *interrupt service routine* secara periodik. Tabel 5.4 berikut adalah implementasi perangkat lunak dari konfigurasi *timer 1* :

**Tabel 5.5 Konfigurasi *Timer 1***

Konfigurasi <i>Timer 1</i>	
1	TCCR1A = 0;
2	TCCR1B = 0;
3	TCNT1 = 59536;
4	TCCR1B  = (1<<CS11);
5	TIMSK1  = (1<<TOIE1);



### 3. Konfigurasi Set Poin

Set poin sangat penting bagi sistem kontroler, karena merupakan titik acuan bagi sistem untuk menentukan sinyal *plant*. Seperti yang telah dijelaskan pada perancangan perangkat lunak bagian set poin, Tabel 5.5 berikut menggambarkan implementasi pada perangkat lunaknya :

**Tabel 5.6 Konfigurasi Set Poin**

Konfigurasi Set Poin	
1	<code>setPoint = 5;</code>

### 4. Meminta Data Sensor

Setelah dikonfigurasi, sensor tidak akan memberikan datanya pada kontroler secara langsung jika tidak diperintahkan oleh kontroler. Sesuai penjelasan pada sub bab 5.1.6 poin nomor 2.1, Tabel 5.6 berikut merupakan implementasi pada perangkat lunaknya :

**Tabel 5.7 Meminta Data Sensor**

Meminta Data Sensor	
1	<code>Wire.beginTransaction(0x68);</code>
2	<code>Wire.write(0x3B);</code>
3	<code>Wire.endTransmission();</code>
4	<code>Wire.requestFrom(0x68,14);</code>

### 5. Menyimpan Data Sensor

Data yang telah *direquest* setelahnya akan segera dikirim oleh sensor, maka dari itu data tersebut harus disimpan untuk nantinya diolah. Tabel 5.7 berikut adalah implementasi perangkat lunak dari menyimpan data sensor :

**Tabel 5.8 Menyimpan Data Sensor**

Menyimpan Data Sensor	
1	<code>rawAX = Wire.read()&lt;&lt;8 Wire.read();</code>
2	<code>rawAY = Wire.read()&lt;&lt;8 Wire.read();</code>
3	<code>rawAZ = Wire.read()&lt;&lt;8 Wire.read();</code>
4	<code>rawTemp = Wire.read()&lt;&lt;8 Wire.read();</code>
5	<code>rawGX = Wire.read()&lt;&lt;8 Wire.read();</code>
6	<code>rawGY = Wire.read()&lt;&lt;8 Wire.read();</code>
7	<code>rawGZ = Wire.read()&lt;&lt;8 Wire.read();</code>

## 6. Mengolah Data Sensor

Data yang diterima oleh sensor tidaklah siap pakai, masih berupa data mentah yang harus dikelola agar bisa dengan mudah dipahami oleh manusia. Tabel 5.8 berikut adalah implementasi perangkat lunak dari pengolahan data sensor :

**Tabel 5.9 Mengelola Data Sensor**

Mengelola Data Sensor	
1	<code>timePrev = times;</code>
2	<code>times = millis();</code>
3	<code>elapsedTime = (times - timePrev) / 1000;</code>
4	<code>acc_total_vector = sqrt((rawAX*rawAX)+(rawAY*rawAY)+(rawAZ*rawAZ));</code>
5	<code>anglePitchAcc = asin(rawAX/acc_total_vector)*57.296;</code>
6	<code>angularYVelocity = rawGY/131;</code>
7	<code>realPitch =</code> <code>0.99*(realPitch+(angularYVelocity*elapsedTime))+0.01*anglePitchAcc;</code>

## 7. Summing Junction

Pada bagian ini adalah inti dari metode *Close Loop System* yang mengurangi set poin dengan nilai sudut bacaan sensor, hasil dari *summing junction* ini akan menjadikan acuan terhadap sinyal *plant* apa yang akan diperintahkan oleh kontroler unit. Tabel 5.9 berikut menggambarkan implementasi perangkat lunak dari *summing junction* :

**Tabel 5.10 Summing Junction**

Summing Junction	
1	<code>Error = setPoint - realPitch;</code>

## 8. Batas Toleransi Error dan Mengubah Nilai Variabel Dir

Batas toleransi error diberikan lantaran pada nilai bacaan sensor didua angka dibelakang koma sangat sensitif terhadap gerakan, maka dari itu untuk membuat aktuator tidak terlalu sering bergerak jika error mendekati nol diberikanla batas toleransi error. Hasil dari error juga menentukan perputaran dari aktuator, Tabel 5.10 berikut adalah implementasi dari batas toleransi error :

**Tabel 5.11 Batas Toleransi Error**

Batas Toleransi Error	
1	<code>if(error&gt;0.05){ dir = 0;</code>
2	<code>} else if(error&lt;-0.05){ dir = 1;</code>
3	<code>} else { dir = 2;</code>
4	<code>}</code>



### 9. Interrupt Service Routine

*ISR* adalah potongan program yang nantinya akan dieksekusi setiap sekian periodik waktu, untuk saat ini sistem diatur agar *ISR* berjalan tiap 3 milidetik. Dalam *ISR* terdapat instruksi – instruksi untuk membaca nilai dari variabel *Dir* dan menentukan perputaran dari aktuator, Tabel 5.11 berikut merupakan potongan program *ISR* :

**Tabel 5.12 Interrupt Service Routine**

Interrupt Service Routine	
1	ISR(TIMER1_OVF_vect) {
2	if(dir == 0){
3	stepCount--;
4	if(stepCount < 0){
5	stepCount = 3;
6	}
7	} else if(dir == 1){
8	stepCount++;
9	if(stepCount > 3){
10	stepCount = 0;
11	}
12	}
13	if(stepCount != stepTemp){
14	switch(stepCount){
15	case 0:
16	digitalWrite(motorA0,HIGH);
17	digitalWrite(motorA1,HIGH);
18	digitalWrite(motorB0,LOW);
19	digitalWrite(motorB1,LOW);
20	break;
21	case 1:
22	digitalWrite(motorA0,LOW);
23	digitalWrite(motorA1,HIGH);
24	digitalWrite(motorB0,HIGH);
25	digitalWrite(motorB1,LOW);
26	break;
27	case 2:
28	digitalWrite(motorA0,LOW);
29	digitalWrite(motorA1,LOW);
30	digitalWrite(motorB0,HIGH);
31	digitalWrite(motorB1,HIGH);
32	break;

Tabel 5.12 (lanjutan)

Interrupt Service Routine	
33	case 3:
34	digitalWrite (motorA0,HIGH);
35	digitalWrite (motorA1,LOW);
36	digitalWrite (motorB0,LOW);
37	digitalWrite (motorB1,HIGH);
38	break;
39	default:
40	digitalWrite (motorA0,LOW);
41	digitalWrite (motorA1,LOW);
42	digitalWrite (motorB0,LOW);
43	digitalWrite (motorB1,LOW);
44	break;
45	}
46	}
47	stepTemp = stepCount;
48	TCNT1 = 59536 ;
49	}

## BAB 6 PENGUJIAN

Pada Bab 6 ini, dilakukan pengujian terhadap sistem yang telah dirancang dan telah diimplementasikan guna mengetahui apakah sistem telah sesuai dengan yang diharapkan dan telah menjadi solusi atas permasalahan yang diangkat atau tidak.

### 6.1. Parameter Pengujian

Parameter yang digunakan untuk menguji sistem ini ada 4 yaitu pembacaan sensor, berat beban (bagian belakang pengemudi) pada sepeda motor, kedua kesalahan sistem dari aktuasi sistem, serta yang ketiga adalah waktu dari proses aktuasi sistem.

### 6.2. Skenario Pengujian

Pengujian kesalahan sistem dan waktu aktuasi dilakukan dengan total 6 orang yang berat badannya berbeda – beda, kemudian dilakukan perekaman pancaran cahaya untuk nantinya dicatat kesalahan sistem dari aktuasi sistem serta lama waktu aktuasi yang dibutuhkan oleh sistem. Tiap 6 orang dilakukan pengujian ulang sebanyak 3 kali sedangkan untuk pengujian pembacaan sensor dilakukan sebanyak 8 kali untuk meminimalisir kesalahan pembacaan oleh mata manusia (peneliti).

Gambar 6.1 berikut menggambarkan skenario pengujian yang dilakukan di lokasi tempat tinggal peneliti, skenario yang dipakai dalam pengujian ini yaitu 1 pengemudi tetap dan 6 penumpang yang berbeda – beda secara bergantian. Terdapat komputer dan kamera digital juga untuk melihat data bacaan sensor serta untuk merekam ketinggian pancaran cahaya.



**Gambar 6.1 Pengujian**

kesalahan sistem dari aktuasi adalah keadaan dimana sistem sudah tidak dapat memberikan pengaruh pada *headlamp* yang bisa dikarenakan torsi aktuator kurang besar atau sudah mencapai batas dari *track* pengatur ketinggian pancaran cahaya. Semakin besar kesalahan sistem maka semakin buruk hasil aktuasi sistem.

Lama waktu aktuasi adalah rentang waktu yang dibutuhkan oleh sistem dari adanya perbedaan sudut *pitch* pada *headlamp* sampai *headlamp* berada pada posisi atau sudut *pitch* yang diinginkan. Waktu yang dibutuhkan tidak ada acuan khusus untuk menentukan bahwa sistem bekerja dengan baik atau tidak, karena ada beberapa hal yang mempengaruhi seperti beban, dan adanya kotoran pada baut pengatur petinggian. Berikut ini adalah macam – macam berat badan orang yang akan diujikan : 60 Kg, 83 Kg, 79 Kg, 77 Kg, 85 Kg, dan 90 Kg.

### 6.3. Pengujian Pembacaan Sensor

Pembacaan sensor adalah hal yang harus diperhatikan mengingat kinerja sistem bergantung pada ketepatan pembacaan sensor. Tata letak dari seluruh bagian pengujian ini sama dengan pada saat konfigurasi set poin pada Bab 5 Perancangan dan Implementasi sub bab 5.1.6. Perancangan Perangkat Lunak poin nomor 1.3. Konfigurasi Set Poin halaman 37. Berikut adalah bagian-bagian dari pengujian ini.

#### 6.3.1. Tujuan Pengujian

Mengetahui kesalahan terhadap pembacaan sensor, meskipun sudah dilakukan pada Bab 5 Sub bab 5.1.5. Perancangan Perangkat Keras : Peletakan Komponen Sistem namun ada baiknya melakukan pengujian ulang untuk mengetahui kesalahannya.

#### 6.3.2. Prosedur Pengujian

Prosedur dalam melakukan pengujian adalah sebagai berikut :

1. Mulai menghidupkan sistem, sepeda motor, dan lampu utama.
2. Mulai menjalankan *Arduino IDE* dan *Serial Monitor* pada komputer.
3. Mulai merekam tampilan layar komputer.
4. Menggerakkan kemudi sepeda motor ke kiri atau ke kanan selama 30 detik agar terjadi kesalahan pembacaan sudut *pitch* (akibat pergantian sudut *pitch* ke *roll* dan sebaliknya) yang berakibat terjadi aktuasi dan merubah sudut *pitch headlamp*, kemudian mengembalikan ke posisi semula (lurus dengan arah sepeda motor).
5. Tunggu hingga selesai melakukan aktuasi, catat tinggi pancaran cahaya pada tembok serta tepuk tangan satu kali sebagai tanda bahwa data tinggi telah diambil.
6. Ulangi langkah 4 dan 5 sebanyak 8 kali lalu hentikan perekaman.
7. Catat nilai bacaan sensor saat terdengar suara tepuk tangan satu kali.

### 6.3.3. Hasil Pengujian

Hasil pengujian berupa video dan informasi tinggi pancaran cahaya pada tembok tanpa adanya beban pada sepeda motor, lalu disajikan dalam bentuk tabel untuk informasi ketinggian yang terjadi setelah aktuasi, data bacaan sensor, sudut bacaan yang seharusnya berdasarkan tinggi pancaran (menggunakan  $\arctan$  sesuai dengan Konfigurasi Set Poin halaman 37) dan kesalahan bacaan sensor. Perlu diingat bahwa penempatan sensor pada *headlamp* adalah terbalik sehingga menghasilkan jarak antara bacaan sensor terhadap sudut *headlamp* yang sebenarnya seperti yang telah dijelaskan pada Bab 5 Perancangan dan Implementasi sub bab 5.1.6. Perancangan Perangkat Lunak poin nomor 1.3. Konfigurasi Set Poin halaman 37. Tabel 6.2 berikut adalah hasil pengujiannya.

**Tabel 6.1 Hasil Pengujian Pembacaan Sensor**

Pecobaan ke -	Tinggi Pancaran Cahaya pada Tembok (cm)	Data Bacaan Sensor (derajat)	Sudut <i>Headlamp</i> berdasarkan Tinggi Pancaran Cahaya pada Tembok (derajat)	Kesalahan Bacaan Sensor
1	73,8	4,96	4.97	0,29%
2	74	5,02	5.02	0,08%
3	74,5	5,01	5.14	2,96%
4	74	5,01	5.02	0,28%
5	74	5	5.02	0,48%
6	74	4,99	5.02	0,68%
7	73,8	5,02	4.97	0,91%
8	73,2	4,87	4.82	0,92%

### 6.3.4. Analisa Pengujian

Data yang di dapat dari hasil pengujian tidak sama antara satu dengan yang lain, hal ini dikarenakan beberapa faktor seperti perubahan jarak antara sepeda motor dengan tembok saat menggerakkan kemudi dan getaran dari mesin sepeda motor. Kolom kesalahan bacaan sensor di dapat dari selisih antara nilai sudut *headlamp* berdasarkan tinggi pancaran cahaya pada tembok dengan nilai data bacaan sensor atau lebih ringkasnya selisih antara sudut yang seharusnya (karena berdasarkan pada tinggi pancaran) dan sudut bacaan sensor lalu dibagi dengan sudut yang seharusnya kemudian dikali 100%.

Rata-rata kesalahan bacaan sensor adalah 0,79% dengan nilai kesalahan maksimal adalah 2,69% dan nilai minimal kesalahan adalah 0,08%, dengan ini di dapat jangkauan kesalahannya sebesar 2,61% yang berarti cukup kecil jangkauan

kesalahannya. Variasi dari hasil pengujian kali ini adalah 0,68 yang berarti hasilnya tidak terlalu bervariasi atau yang berarti hasilnya mirip-mirip. Standar deviasi dari hasil pengujian ini juga terbilang kecil yaitu 0,82 yang berarti simpangan antara nilai tiap pengujian dengan nilai rata-ratanya tidak besar dan bahkan hampir sama, dengan begitu maka nilai rata-rata dari pengujian ini mewakili pengujian kali ini.

#### 6.4. Pengujian Kesalahan Sistem

Bagian pertama dari pengujian, kesalahan sistem dapat menyebabkan perubahan jangkauan cahaya dari yang telah dirancang. Pada sub bab berikut akan menjelaskan tujuan, prosedur, hasil, dan analisa pengujian.

##### 6.4.1. Tujuan Pengujian

Pengujian kesalahan sistem ditujukan untuk mengetahui tingkat keberhasilan sistem dalam melaksanakan tugasnya. Tingkat keberhasilannya adalah sistem dapat memberikan aktuasi pada *headlamp* berupa mengatur ketinggian pancaran cahaya pada tembok setinggi 74 cm sesuai dengan perancangan set poin. Angka 74 cm didapat dari Persamaan 5.24 yang dijelaskan pada Perancangan Perangkat Lunak halaman 40.

##### 6.4.2. Prosedur Pengujian

Prosedur dalam melakukan pengujian adalah sebagai berikut :

1. Pengemudi menaiki sepeda motor.
2. Kamera perekam ketinggian pancaran cahaya pada tembok diaktifkan dan merekam.
3. Lampu utama sepeda motor dihidupkan, kondisi *Low Beam* atau lampu dekat.
4. Komputer menyala dan menunjukkan nilai sudut bacaan sensor.
5. Tunggu hingga sudut bacaan sensor sampai 5°.
6. Penumpang menaiki sepeda motor.
7. Tunggu hingga tidak ada perubahan ketinggian pancaran cahaya pada tembok.
8. Penumpang turun.
9. Kamera perekam berhenti merekam.

##### 6.4.3. Hasil Pengujian

Hasil pengujian berupa video dan dilihat satu persatu untuk semua penumpang lalu disajikan dalam bentuk tabel untuk ketinggian pancaran cahaya pada tembok serta kesalahan sistem yang terjadi. Tabel 6.1 berikut adalah hasil pengujiannya.



**Tabel 6.2 Hasil Pengujian Kesalahan Sistem pada Permukaan Datar**

Pecobaan ke -	Berat Badan Penumpang (Kg)	Ketentuan Tinggi Pancaran Cahaya pada Tembok (cm)	Tinggi Pancaran Cahaya pada Tembok (cm)	Kesalahan (%)
1	60	74	74	0 %
2	60	74	74,1	0,13%
3	60	74	74	0%
4	83	74	73,5	0,67 %
5	83	74	74,2	0,27%
6	83	74	74,1	0,13%
7	79	74	72,2	2,43 %
8	79	74	74	0%
9	79	74	74,1	0,13%
10	77	74	75,5	2,027 %
11	77	74	74,2	0,27%
12	77	74	74	0%
13	85	74	75,9	2,567 %
14	85	74	74,5	0,67%
15	85	74	74,1	0,13%
16	90	74	76,5	3,378 %
17	90	74	74,3	0,40%
18	90	74	74	0%

#### 6.4.4. Analisa Hasil Pengujian

Dari 18 kali percobaan dengan berat badan penumpang yang berbeda didapatkan hasil kesalahan sistem yang berbeda-beda, rata-rata kesalahan sistem dari 18 percobaan adalah 0,73%. Untuk rata-rata kesalahan sistem tiap orang adalah 0,04% untuk berat badan 60 Kg, 0,35% untuk berat badan 83 Kg, 0,85% untuk berat badan 79 Kg, 0,76% untuk berat badan 77 Kg, 1,12% untuk berat badan 85 Kg, dan 1,25% untuk berat badan 90 Kg.

Berat badan 90 Kg menjadi pengujian dengan kesalahan sistem paling besar yaitu 1,25%, hal ini di akibatkan karena semakin berat beban di bagian belakang pengemudi maka sudut *pitch* dari sepeda motor akan semakin membesar dan membuat aktuator harus lebih ekstra menurunkan atau menekan *headlamp*

kebelakang guna menurunkan pancaran sinar. Pada penjelasan rata-rata kesalahan sistem tiap orang juga memberikan hasil bahwa semakin berat beban yang ada pada bagian belakang pengemudi maka kesalahan sistem juga cenderung lebih besar.

Jika di lihat dari percobaan yang mengalami kesalahan sistem, tinggi pancarannya pada tembok ada yang kurang dari 74 cm dan lebih dari 74 cm yang berarti lebih rendah dan lebih tinggi dari set poin. Hal ini akan mengakibatkan jangkauan pancaran cahaya ada yang lebih pendek dari perancangan dan ada yang lebih panjang dari perancangan, dan kejadian ini dimungkinkan adanya perubahan jarak antara sumber cahaya dengan tembok saat penumpang naik.

Nilai kesalahan sistem paling besar adalah 3,378% dan paling kecil adalah 0%, maka dari itu didapat jangkauan kesalahannya adalah 3,378%. Variasi dari 18 pengujian kesalahan sistem adalah 1,15 yang membuktikan bahwa kesalahan sistem tidak terlalu bervariasi atau hampir mirip antar hasil. Selain itu hasil perhitungan Standar Deviasi dari 18 pengujian kesalahan sistem adalah 1,07 yang membuktikan bahwa simpangan antara tiap data terhadap rata-ratanya tidak terlalu besar, dengan ini maka data rata-rata pengujian kesalahan sistem dapat dijadikan hasil atau dapat mewakili hasil dari 18 kali percobaan pada pengujian kesalahan sistem ini.

## **6.5. Pengujian Waktu**

Waktu adalah salah satu faktor yang di pengaruhi oleh perbedaan berat badan penumpang, berikut ini adalah bagian – bagian dari pengujian waktu aktuasi sistem.

### **6.5.1. Tujuan Pengujian**

Mengetahui waktu yang dibutuhkan oleh sistem dalam melakukan pengaturan ketinggian pancaran cahaya, tidak ada aturan khusus dari pemerintah untuk batasan waktu pengaturan ketinggian lampu sehingga tidak ada kesalahan sistem pada pengujian kali ini.

### **6.5.2. Prosedur Pengujian**

Prosedur dalam melakukan pengujian adalah sebagai berikut :

1. Pengemudi menaiki sepeda motor.
2. Kamera perekam ketinggian pancarah cahaya pada tembok diaktifkan dan merekam.
3. Lampu utama sepeda motor dihidupkan, kondisi *Low Beam* atau lampu dekat.
4. Komputer menyala dan menunjukkan nilai sudut bacaan sensor.
5. Tunggu hingga sudut bacaan sensor sampai 5°.
6. Penumpang menaiki sepeda motor.

7. Tunggu hingga tidak ada perubahan ketinggian pancaran cahaya pada tembok.
8. Penumpang turun.
9. Kamera perekam berhenti merekam.

### 6.5.3. Hasil Pengujian

Hasil pengujian berupa video dan dilihat satu persatu untuk semua penumpang lalu disajikan dalam bentuk tabel untuk waktu aktuasi sistem dalam mengatur ketinggian pancaran cahaya pada tembok. Tabel 6.2 berikut adalah hasil pengujiannya.

**Tabel 6.3 Hasil Pengujian Waktu pada Permukaan Datar**

Pecobaan ke -	Berat Badan Penumpang (Kg)	Waktu (detik)
1	60	35
2	60	34
3	60	35
4	83	54
5	83	52
6	83	53
7	79	56
8	79	55
9	79	56
10	77	36
11	77	45
12	77	47
13	85	40
14	85	55
15	85	53
16	90	40
17	90	57
18	90	60

### 6.5.4. Analisa Pengujian

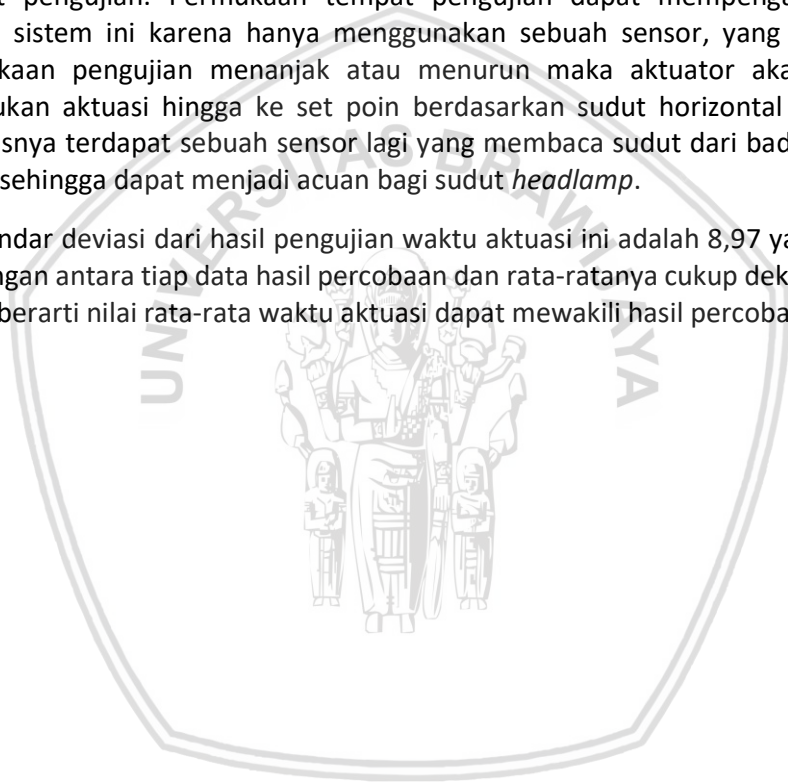
Dari hasil pengujian yang berupa waktu aktuasi, terlihat bahwa semakin bertambahnya berat badan penumpang maka waktu aktuasi. Hal ini dikarenakan

saat bertambahnya beban maka sudut *pitch* yang berubah semakin besar dan membutuhkan aktuasi lebih lama untuk menyesuaikan dengan set poin. Namun dapat diperoleh rata-rata selama 47,9 detik.

Terdapat beberapa analisis yang dapat dilakukan seperti nilai maksimal dari waktu aktuasi sebesar 60 detik dan nilai minimal dari waktu aktuasi sebesar 34 detik sehingga didapat jangkauan sebesar 26 detik yang berarti untuk rantang berat badan 60 Kg sampai 90 Kg atau jangkauan berat badan 30 Kg, perubahan waktu aktuasi berada dikisaran 26 detik.

Variasi dari hasil pengujian adalah sebesar 80,52 yang berarti hasil dari aktuasi cukup beragam atau cukup bervariasi, hal ini terjadi karena faktor permukaan tempat pengujian. Permukaan tempat pengujian dapat mempengaruhi hasil karena sistem ini karena hanya menggunakan sebuah sensor, yang mana jika permukaan pengujian menanjak atau menurun maka aktuator akan bekerja melakukan aktuasi hingga ke set poin berdasarkan sudut horizontal 0 derajat. Seharusnya terdapat sebuah sensor lagi yang membaca sudut dari badan sepeda motor sehingga dapat menjadi acuan bagi sudut *headlamp*.

Standar deviasi dari hasil pengujian waktu aktuasi ini adalah 8,97 yang berarti simpangan antara tiap data hasil percobaan dan rata-ratanya cukup dekat, dengan hal ini berarti nilai rata-rata waktu aktuasi dapat mewakili hasil percobaan ini.



## BAB 7 PENUTUP

Setelah pengujian dilakukan berikutnya adalah penarikan kesimpulan berdasarkan segala aspek dari penelitian ini mulai dari perancangan, implementasi, hingga hasil pengujian. Tidak lupa juga saran dari peneliti untuk dapat mengembangkan dan/atau memperbaiki sistem ini yang akan dijelaskan pada sub bab dibawah ini.

### 7.1. Kesimpulan

Berdasarkan rumusan masalah, perancangan, implementasi, hingga hasil pengujian, peneliti menyimpulkan :

1. Sistem dapat mengetahui sudut *pitch* saat ini dengan menggunakan sensor *Inertia Measuring Unit* MPU6050 yang didalamnya telah terdapat *Accelerometer* dan *Gyroscope*. Sistem menggunakan 3 sudut dari *Accelerometer* serta 3 sudut dari *Gyroscope* yang berarti total menggunakan 6 sudut dari sensor. Pada pengujian pertama tentang pengujian pembacaan sensor juga membuktikan bahwa sistem dapat mengetahui sudut *pitch* saat ini dengan bantuan sensor *IMU* MPU6050.
2. Sistem dapat mengontrol nilai sudut *pitch* berkat sistem aktuator berupa *stepper* motor dan mekanismenya yang mempengaruhi *headlamp* serta metode *Close Loop System* sebagai pengontrolnya. Hal ini berkaitan dengan pengujian kesalahan sistem dan waktu aktuasi pada Bab 6 yang membuktikan bahwa sistem dapat mengontrol sudut *pitch headlamp* karena adanya sistem aktuator dan mekanismenya beserta metode pengontrolnya.
3. Implementasi metode *Close Loop System* ada 2 bagian yaitu perangkat lunak dan perangkat keras. Pada perangkat lunak, implementasi metode diharuskan melakukan penerimaan data umpan balik dari sensor yang bergantung pada aktuator. Sedangkan pada perangkat keras, objek yang dikontrol haruslah dapat dibaca sudutnya oleh sensor serta objek tersebut harus dapat dikontrol oleh aktuator. Sehingga setiap kali sistem melakukan aktuasi, maka objek yang dikontrol akan terpengaruh dan pengaruhnya itu dibaca oleh sensor yang menjadikannya data umpan balik bagi sistem.
4. Dengan melihat tinggi pancaran pada tembok (kolom nomor 4 pada Tabel 6.2) dapat diketahui kesalahan dari aktuasi sistem yang juga menunjukkan performansi sistem. Performansi dari sistem dengan pengontrol *Close Loop System* sangat bergantung pada aktuator, permukaan saat melakukan aktuasi, serta *noise* dari mesin sepeda motor, jika aktuator terlalu sigap kemungkinan hal yang akan terjadi adalah hasil kontrol yang berosilasi, jika permukaan menaik atau menurun maka sistem akan menganggap bahwa sepeda motor berada pada posisi datar, dan *noise* dari mesin akan mempengaruhi *accelerometer* yang juga dapat mempengaruhi hasil dari bacaan sudut *pitch*.

Kesalahan sistem rata-rata adalah 0,73 %, pembacaan sensor mempunyai rata-rata kesalahan sebesar 0,79% serta lama waktu aktuasi rata-rata sebesar 47,9 detik.

Untuk waktu yang dibutuhkan dalam melakukan aktuasi beragam–ragam sesuai dengan berat badan penumpang, namun dengan rata-rata selama 47,9 detik atau dapat dibulatkan menjadi 60 detik pun akan terasa lebih cepat dibandingkan dengan pengaturan manual menggunakan engkol.

## 7.2. Saran

Selanjutnya adalah saran dari peneliti terhadap pengembangan selanjutnya pada sistem pengontrol ketinggian pancaran cahaya ini :

1. Paling penting adalah aktuator pada sistem yang menentukan tingkat keberhasilan sistem, pada pengembangan selanjutnya menggunakan aktuator dengan torsi lebih besar seperti contoh Stepper Motor Nema 17 akan lebih meningkatkan tingkat keberhasilan sistem.
2. Kontroler unit dibuat dengan hanya 1 PCB yang menjadi satu dengan PSU akan membuatnya lebih ringkas sehingga dapat dengan mudah diberi tempat khusus yang dapat membuatnya lebih tahan terhadap segala kondisi cuaca.
3. Penggunaan 2 buah sensor untuk dapat mendeteksi perubahan sudut *pitch* dalam keadaan sepeda motor bergerak akan lebih mempermudah pengemudi.
4. Keluaran berupa informasi sudut *pitch* yang ditampilkan dalam LCD atau *Seven Segment* akan memberikan rasa percaya diri pada pengemudi.
5. Tambahan *override mode* sehingga sistem dapat bekerja secara otomatis atau manual, namun masih dalam bentuk digital sehingga pengemudi hanya perlu menekan tombol untuk menaikkan dan menurunkan pancaran cahaya.



## DAFTAR PUSTAKA

- AAAFoundation, 2012. How To Avoid Headlight Glare. [pdf] Northwest, Washington, United States of America. Tersedia di: <[https://mafiadoc.com/how-to-avoid-headlight-glare-aaa-foundation\\_59c530411723dd2c1ca9e647.html](https://mafiadoc.com/how-to-avoid-headlight-glare-aaa-foundation_59c530411723dd2c1ca9e647.html)> [Diakses 26 Februari 2018]
- Colton, S. 2007. The Balance Filter. [online] Tersedia di: < Northwest, Washington, United States of America > [Diakses 3 Maret 2018]
- Dastern.torque.net, 2014. Hella Headlamp System. [online] Tersedia di: <[http://dastern.torque.net/techdocs/Archives/Hella\\_Systems/de.htm](http://dastern.torque.net/techdocs/Archives/Hella_Systems/de.htm)> [Diakses 5 Februari 2018]
- Debra, 2018. Gyroscopes and Accelerometers on a Chip. [online] Tersedia di: <<http://www.geekmomprojects.com/gyroscopes-and-accelerometers-on-a-chip/>> [Diakses 3 Maret 2018]
- Electronics, H. 2018. Hobby Electronics. [online] Tersedia di: <<http://www.hobbyelectronics.co.nz/product.php?SKU=141>> [Diakses 15 Februari 2018]
- Electronicshub, 2015. Closed Loop System. [online] Tersedia di: <<https://www.electronicshub.org/closed-loop-system/>> [Diakses 4 Maret 2018]
- Geekbuying, 2018. GY-521 MPU-6050 Module Three Axis Accelerometer Sensor Module For Arduino Raspberry Pi AVR ARM. [online] Tersedia di: <<https://www.geekbuying.com/item/GY-521-MPU-6050-Module-Three-Axis-Accelerometer-Sensor-Module-For-Arduino-Raspberry-Pi-AVR-ARM-344582.html>> [Diakses 8 Januari 2018]
- Indonesia, 2012. Peraturan Pemerintah Republik Indonesia Nomor 55 Tahun 2012 Tentang Kendaraan. [pdf] DKI Jakarta, Indonesia. Tersedia di: <<http://hubdat.dephub.go.id/peraturan-pemerintah/1421-peraturan-pemerintah-republik-indonesia-nomor-55-tahun-2012-tentang-kendaraan>> [Diakses 8 Januari 2018]
- Instruments, T. 2018. Peripheral Driver and Actuator. [pdf] Tersedia di: <<http://www.ti.com/product/L293D>> [Diakses 15 Februari 2018]
- Instruments, N. 2011. PID Theory Explained. [online] Tersedia di: <<http://www.ni.com/white-paper/3782/en/#toc2>> [Diakses 20 Februari 2018]
- Invensense, 2017. MPU6050. [pdf] Tersedia di: <<https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>> [Diakses 8 Januari 2018]

- Mraz, S. 2014. Mechanic Design. [online] Tersedia di: <<http://www.machinedesign.com/engineering-essentials/what-s-difference-between-pitch-roll-and-yaw>> [Diakses 8 Januari 2018]
- Nasir, 2014. Difference Between Open Loop & Closed Loop Systems. Tersedia di: <<http://engineering.electrical-equipment.org/panel-building/difference-between-open-loop-closed-loop-systems.html>> [Diakses 7 Januari 2018]
- Peter Opgen-Rhein, T. B. 2004. A HARDWARE-IN-THE-LOOP BASED PROCESS IMPROVE QUALITY AND DECREASES THE DEVELOPMENT PERIOD. [pdf] Tersedia di: <[https://www.researchgate.net/publication/316904354\\_A\\_Hardware\\_in-the-Loop\\_Based\\_Process\\_Improves\\_Quality\\_and\\_Decreases\\_the\\_Development\\_Period\\_of\\_a\\_Dynamic\\_Headlamp\\_Levelling\\_System](https://www.researchgate.net/publication/316904354_A_Hardware_in-the-Loop_Based_Process_Improves_Quality_and_Decreases_the_Development_Period_of_a_Dynamic_Headlamp_Levelling_System)> [Diakses 7 Januari 2018]
- Polantas, 2017. Kecelakaan di Indonesia Selama Triwulan Terakhir. [online] Tersedia di: <<http://www.korlantas-irsms.info/graph/accidentData>> [Diakses 18 Januari 2018]
- S. Karaman, T. H. 2006. DESIGN AND HARDWARE-IN-THE-LOOP SIMULATION OF AN AUTOMATIC HEADLIGHT. [pdf] Tersedia di: < [https://ac.els-cdn.com/S1474667015342579/1-s2.0-S1474667015342579-main.pdf?\\_tid=7950ce3b-316c-4a8d-a5e7-715a7d410ce7&acdnat=1526490846\\_4fb8d9e7fe357629608348cb0f9f7577](https://ac.els-cdn.com/S1474667015342579/1-s2.0-S1474667015342579-main.pdf?_tid=7950ce3b-316c-4a8d-a5e7-715a7d410ce7&acdnat=1526490846_4fb8d9e7fe357629608348cb0f9f7577)> [Diakses 7 Januari 2018]
- Stateham, B. 2016. The Maker Show: Episode 8 - Driving Your Stepper Motor with an Arduino. [online] Tersedia di: <<https://channel9.msdn.com/Shows/themakershow/8>> [Diakses 14 April 2018]
- Store.Arduino. 2018. Arduino Nano. Tersedia di: <<https://store.arduino.cc/usa/arduino-nano>> [Diakses 7 Januari 2018]
- UNECE. 2014. UN Vehicle Regulations - 1958 Agreement. [pdf] Tersedia di: <<https://www.unece.org/fileadmin/DAM/trans/main/wp29/wp29regs/2017/E-ECE-TRANS-505-Rev.3e.pdf>> [Diakses 7 Januari 2018]